ePassport:
Side Channel in the Basic Access Control

Luigi Sportiello(✉)

European Commission, Joint Research Centre,
Via Enrico Fermi 2749, 21027 Ispra, VA, Italy
luigi.sportiello@jrc.ec.europa.eu

Abstract. An electronic version of the traditional passport (ePassport) is nowadays issued by many countries to their citizens. A contactless chip storing personal details of the document holder is embedded in the ePassport cover. To prevent unauthorized reads of the chip's content and to protect its communication with a legitimate reader the Basic Access Control (BAC) has been introduced. Thanks to the BAC, only those readers aware of the secret associated with an ePassport chip can access its content. In this paper we show that a side channel analysis can be carried out for some chips secured with the BAC. In particular we analyze the chip response time during BAC operations, showing how the collected data could be exploited to mount an attack in order to get access to the chip's content. We have verified the presence of such side channel in real ePassports and stress that electronic Driving Licences could be affected as well, since the same access control mechanism is adopted for them.

Keywords: ePassport · Basic access control · Side channel analysis · eDriving licence

1 Introduction

Nowadays many countries all over the world issue the electronic version of the passport (ePassport) [1,2] to their citizens, the international document used by people for their identification abroad. In contrast to the traditional passport, in the ePassport a chip is embedded in the cover of the document. Such electronic component stores personal data regarding the document holder and can be accessed by a contactless interface: a reader put in proximity of the document, following the RFID communication principle, powers the chip and exchanges messages with it. Due to the sensitivity of the involved data and the over-the-air nature of the communication, a mechanism to protect the access to the chip and the communication with it has been introduced, the Basic Access Control (BAC). The BAC is a mutual authentication protocol between chip and reader based on symmetric-key cryptography: every ePassport is featured by a secret string called Machine Readable Zone (MRZ) and a reader has to be aware of

© European Union 2014
N. Saxena and A.-R. Sadeghi (Eds.): RFIDSec 2014, LNCS 8651, pp. 173–184, 2014.
DOI: 10.1007/978-3-319-13066-8_11

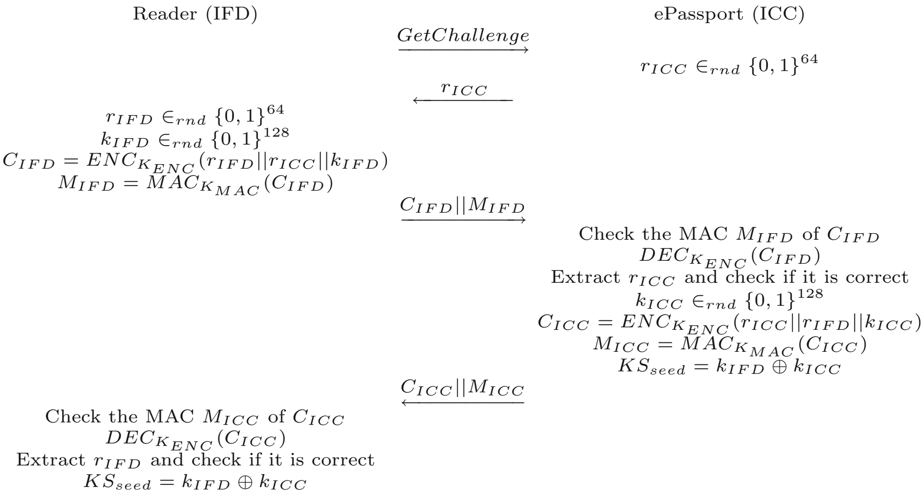

Reader (IFD)
ePassport (ICC)
$\xrightarrow{GetChallenge}$
$r_{ICC} \in_{rnd} \{0,1\}^{64}$
$\xleftarrow{r_{ICC}}$
$r_{IFD} \in_{rnd} \{0,1\}^{64}$
$k_{IFD} \in_{rnd} \{0,1\}^{128}$
$C_{IFD} = ENC_{K_{ENC}}(r_{IFD}||r_{ICC}||k_{IFD})$
$M_{IFD} = MAC_{K_{MAC}}(C_{IFD})$
$\xrightarrow{C_{IFD}||M_{IFD}}$
Check the MAC M_{IFD} of C_{IFD}
$DEC_{K_{ENC}}(C_{IFD})$
Extract r_{ICC} and check if it is correct
$k_{ICC} \in_{rnd} \{0,1\}^{128}$
$C_{ICC} = ENC_{K_{ENC}}(r_{ICC}||r_{IFD}||k_{ICC})$
$M_{ICC} = MAC_{K_{MAC}}(C_{ICC})$
$KS_{seed} = k_{IFD} \oplus k_{ICC}$
$\xleftarrow{C_{ICC}||M_{ICC}}$
Check the MAC M_{ICC} of C_{ICC}
$DEC_{K_{ENC}}(C_{ICC})$
Extract r_{IFD} and check if it is correct
$KS_{seed} = k_{IFD} \oplus k_{ICC}$

Fig. 1. Basic Access Control between Reader (also known as InterFace Device - IFD) and ePassport chip (also known as Integrated Circuit Card - ICC). ENC/DEC represent a cipher based on Triple-DES in CBC mode with zero IV, while MAC generates a 8-byte message authentication code according to the ISO/IEC 9797-1 MAC Algorithm 3 [2].

such string to successfully run the BAC with the document's chip and establish a communication with it. At the end of the BAC a couple of secret keys are agreed by the two parties and the following communication is then encrypted and authenticated.

In the literature different works highlighting some weaknesses of the BAC have been published. In particular, the majority of them reports that the entropy associated to the MRZ is quite low putting at stake the data stored in the chip. This has been pointed out for Belgian, Dutch, German, American and Italian ePassports [3–7]. In addition, the ePassports issued by a country over time could be featured by different chip versions, each associated to a subset of possible MRZs, so querying properly an ePassport chip its version can be identified associating it to a quite low entropy set of MRZs [7]. The authors of [6] show that in case of low entropy MRZs specific cracking machines can be used to attack a recorded BAC communication between a reader and an ePassport trying to get the relative MRZ.

In this paper we show that the BAC security could be also affected by the analysis of side channels. In cryptography the side channel analysis denotes the examination of information unintentionally leaked by a device regarding its internal execution of a cryptographic operation. Such analysis can be exploited to breach the security of cryptographic mechanisms. We have found out that timing analysis can be carried out during the execution of the BAC for some ePassport implementations. The first example of timing analysis against ePassports is reported in [8], where the authors analyzing the response times of ePassport chips were able to track them without breaking the relative cryptographic protocol. In our work we examine the response times of ePassports solicited with specific pre-formatted commands and show how such analysis, when combined with the MRZ low entropy issue, could be used to mount an attack against the BAC if no countermeasures are taken. We have detected the side channel in a subset of the examined ePassports, as it basically depends on the specific implementation of the adopted cryptographic algorithms in the chip, but for all of them a countermeasure was able to prevent the attack designed in this paper to exploit such timing side channel. We also point out that for such attack to be successful, an interaction of several days with the document's chip would be required, and this may be hard to achieve in practice. Nevertheless, the detected side channel analysis allows to retrieve data that the chip is not suppose to leak, that being so a security assessment of the current ePassport implementations is advisable.

The paper is organized as follows. Section 2 introduces the BAC. In Sect. 3 we present the side channel found out during the execution of the BAC in some ePassport implementations, while in Sect. 4 we show how it could be exploited to set up an attack against ePassport chips in case no specific countermeasures are adopted. We discuss some implications of our work in Sect. 5 and give conclusions in Sect. 6.

2 Basic Access Control

The BAC has been introduced to prevent the unauthorized read of the ePassport's chip content and to guarantee confidentiality during the communication with a reader. It is a mutual authentication protocol based on a common secret shared by a chip (Integrated Circuit Card - ICC) and a reader (InterFace Device - IFD) that intend to communicate together. Such secret is represented by a string, called Machine Readable Zone (MRZ), printed in an internal data page of the ePassport. The idea behind the MRZ is that only the ePassport holder can authorize the access to the chip of his document explicitly showing such page: the string can then be optically scanned or typed by an operator and given to the reader. Each ePassport is featured by a unique MRZ and its typical form is the following

```
P<UTOSURNAME<<NAME<<<<<<<<<<<<<<<<<<<<<<<<<<
1234567897UTO6908061F9406236<<<<<<<<<<<<<<04
```

where the information encoded in the second line, the only one used for the BAC, is the following: 9 characters representing the passport number (PN) followed by a check digit, 3 characters reserved for the nationality, 6 characters for the date of birth (DB) followed by a check digit, one character representing the gender of the holder, 6 characters for the document expiration date (DE) and in the end padding symbols (<) followed by two final check digits (the check digits are defined in [1]). The MRZ is used by the two parties to derive a key pair, K_{ENC} and K_{MAC}, as follows

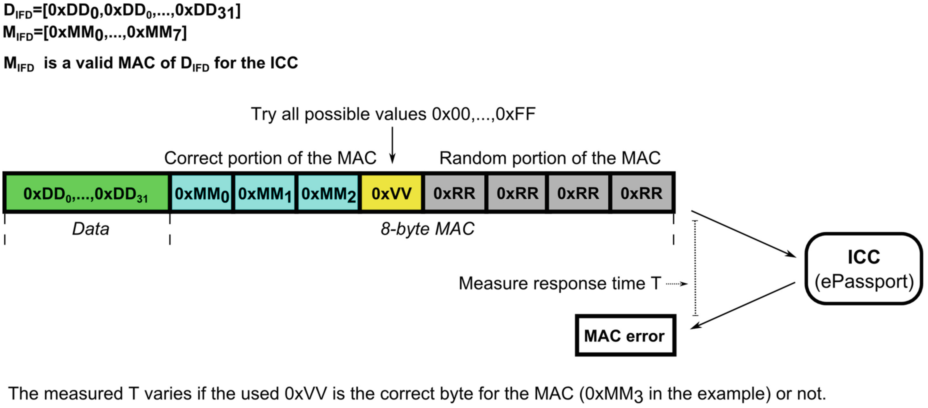

D_IFD=[0xDD0,0xDD0,...,0xDD31]
M_IFD=[0xMM0,...,0xMM7]
M_IFD is a valid MAC of D_IFD for the ICC
Try all possible values 0x00,...,0xFF
Correct portion of the MAC
Random portion of the MAC
0xDD0,...,0xDD31
0xMM0
0xMM1
0xMM2
0xVV
0xRR
0xRR
0xRR
0xRR
Data
8-byte MAC
Measure response time T
ICC (ePassport)
MAC error
The measured T varies if the used 0xVV is the correct byte for the MAC (0xMM3 in the example) or not.

Fig. 2. ePassport MAC checking: timing analysis of a specific MAC byte.

$$\begin{gathered}
\text{MRZ\_information}=\text{PN}||\text{DB}||\text{DE} \\
K_{seed}=\text{msb\_16}(\text{SHA-1}(\text{MRZ\_information})) \\
K_{ENC}=\text{msb\_16}(\text{SHA-1}(K_{seed}||00000001)) \\
K_{MAC}=\text{msb\_16}(\text{SHA-1}(K_{seed}||00000002))
\end{gathered}$$

where PN, DB and DE are respectively followed by their check digit and msb_N stands for “the N most significant bytes”.

The full BAC protocol is shown in Fig. 1. K_{ENC} and K_{MAC} are respectively used to encrypt the data exchanged by the parties, relying on a Triple-DES in CBC mode with zero IV, and to compute a relative 8-byte message authentication code (MAC) according to the ISO/IEC 9797-1 MAC Algorithm 3 [2]. At the beginning the ICC, solicited by the IFD, generates and sends a random value. Such value is encrypted by the IFD along with an additional pair of generated random numbers obtaining C_{IFD}, which is sent to the ICC together with its MAC M_{IFD}. Firstly the ICC checks the MAC and, if valid, decrypts C_{IFD} verifying that the random number generated at the beginning is correctly returned. If so, the protocol continues in reverse order, with further random material generated by the ICC, which is encrypted, authenticated and sent to the IFD. If also the IFD checks are successful, the random values exchanged by the parties are used to set up a common secret KS_{seed}. From this agreed secret a session key pair is generated to encrypt and authenticate the following communication.

3 Side Channel Discovery

We focused our attention on the message $C_{IFD}||M_{IFD}$ sent by the reader to the chip during the BAC (Fig. 1). Upon the arrival of the message, the chip has to perform some checks on it and in case of failure an error message is returned. Measuring the response time of such error messages it is possible to extract some information concerning the internal checks carried out by the chip.

Given:
K_{MAC} = Valid MAC key for the ICC
$D_{IFD}[32] = [0xDD_0, \ldots, 0xDD_{31}]$ //Generic 32-byte vector
$M_{IFD} = MAC_{K_{MAC}}(D_{IFD})$ //Valid authentication code of D_{IFD} for the ICC

Consider the MAC sent by the IFD to the ICC during the BAC as an 8-byte vector. For each byte T ($T \in [0, 7]$) collect time statistics running $ByteStatistics(D_{IFD}, M_{IFD}, T)$.

```
$ByteStatistics(D_{IFD}, M_{IFD}, T)$:
$T$ = MAC target byte to be analyzed
$MeanResponseTimes[256]$;

$for(V = 0x00\ to\ 0xFF)$ : //Try all possible values $V$ in the byte $T$ of the MAC
    $MeanResponseTime[V] = ByteValueStatistics(D_{IFD}, M_{IFD}, T, V)$;

Plot $MeanResponseTimes$;
```

$ByteValueStatistics(D_{IFD}, M_{IFD}, T, V)$:

Reader (IFD)	–	ePassport (ICC)
T = MAC target byte $V = 0xVV$ //Value for the target byte $M_{IFD}[8] = [0xMM_0, \ldots, 0xMM_7]$		
$TestMAC = [0xMM_0, \ldots, 0xMM_{T-1}, 0xVV,$ $0xRR_{T+1}, \ldots, 0xRR_7]$ //$0xRR_i$ is a random byte		
	$\xrightarrow{GetChallenge}$	
	$\xleftarrow{r_{ICC}}$	
Take time T_1		
	$\xrightarrow{D_{IFD}\|\|TestMAC}$	
	$\xleftarrow{error}$	
Take time T_2 Record response time $T_2 - T_1$		
Routine repeated N times returning the mean of the recorded response times		

Fig. 3. Our timing analysis on the MAC checking of a given ePassport.

For instance, for some ePassport implementations, as already highlighted in [8], the error response time differs if there is an immediate failure during the MAC check or later during the verification of the returned random number r_{ICC}: the second case takes longer as also the C_{IFD} decryption is performed, while in the first case such computation is simply skipped.

We decided to perform a deeper timing analysis on the MAC check failure. Our idea is shown in Fig. 2. For a given chip we prepare messages of the form $C_{IFD}||M_{IFD}$ with a partially erroneous 8-byte MAC, which is made of a series of correct bytes, followed by a target byte, appending in the end random bytes. The value of the so-called target byte is varied among all possible byte values, so 256 messages are generated for a given target byte. Such messages are sent to the chip, which will reply with MAC error messages, and the relative response times are measured. The response time should differ when the right value, considering the valid MAC that should be attached to the message, is used in the target byte.

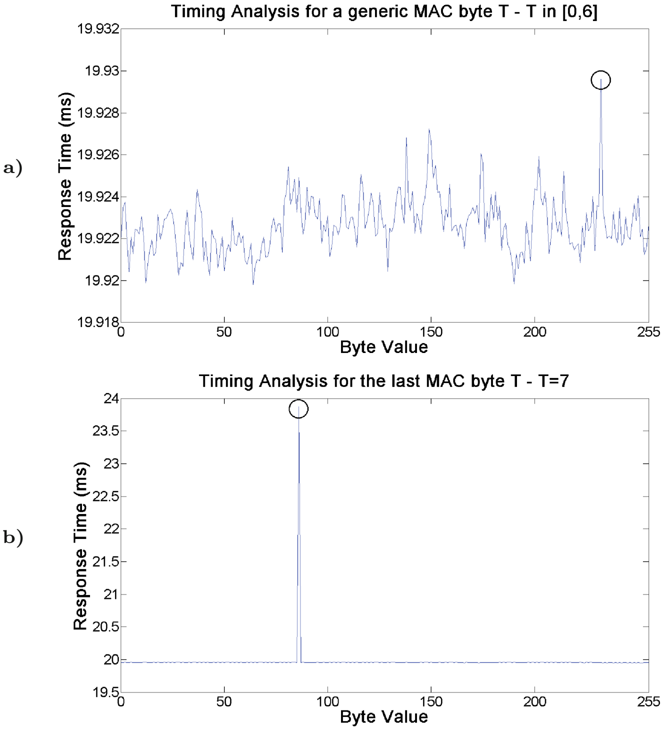


Fig. 4. MAC bytes value identification through MAC checking response times. A representative chart of the timing analysis on a generic byte of the MAC (excluded the last one) is presented in figure *a*: varying the byte value the MAC checking response time (mean on 5000 iterations) is affected and the highest peak identifies the correct byte value. In figure *b* a representative timing analysis (mean on 500 iterations) for the last byte of the MAC is shown: an outstanding peak clearly identifies the right byte value.

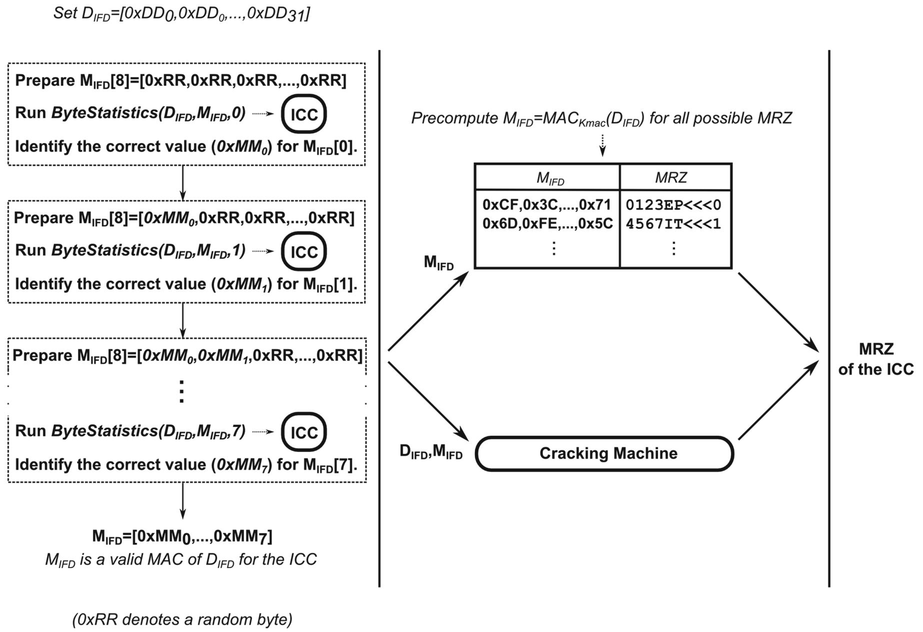
Note that to highlight the time difference a single attempt for each message is not sufficient and a statistics has to be created: each message is sent N times averaging its response times. Our full timing analysis considering in turn all the MAC bytes as target is summarized in Fig. 3.

We have successfully tested our analysis on four Italian ePassports issued between 2009 and 2010. We relied on the libnfc library [9] to develop our timing analysis software that was run on an Ubuntu machine, using an ACS ACR122 contactless reader to interrogate the chips. In our experiments, for a given ePassport with its MRZ, we set $D_{IFD} = ENC_{K_{ENC}}(0x00_0, \ldots, 0x00_{31})$ at the beginning of our timing analysis presented in Fig. 3. Two examples of timing analysis charts are presented in Fig 4. For a given target byte T of the MAC, a chart shows the average response time of the MAC checking operation performed by the chip varying the value of such target byte. We point out that a trimmed mean has been used to compute the average response time associated with each byte value, discarding 5 % of the measures equally distributed between lowest and highest times, to make the charts clearer and less affected by outliers.

In our experiments the timing analysis for each of the first seven bytes of the MAC of our examined ePassports presented similar results and a representative example based on $N = 5000$ iterations is given in the chart of Fig. 4a: some peaks are present with the highest one that identifies the right value for the byte under examination. We have verified that for a lower number of iterations the peak associated with the right value could not emerge remaining immersed at same level of others, so without revealing the correct byte value. In that regard, we have verified that in general a higher number of iterations allows to achieve more reliable analysis. The chart of Fig. 4b shows a representative timing analysis, based on $N = 500$ iterations, for the last byte of a MAC: the peak identifying the right byte value is clearly evident. We also ran our analysis on two Italian ePassports dating back to 2007, but for them no peaks appeared (apart from the last byte), so our analysis was not apparently applicable against them.

It is difficult to give an explanation for the chip behaviour regarding the first seven bytes of the MAC, because the details regarding the internal implemented solutions are not publicly released and we have to look at the chip as a black box. Despite that, according to the results, a tight relation with the specific MAC checking algorithm implementation in the chip seems evident. Indeed, the timing analysis is effective for a subset of documents but not for others issued in a different period of time, and as stated in [7] different versions of ePassport chips are issued by the country over time, each one probably featured by a specific hardware platform and ePassport software. We suppose that for the affected version the MAC value is somehow internally checked in a byte by byte manner starting from the first one and when a wrong byte is found some decision is taken. In that regard, we also point out that some tests sending to the chip a MAC with the correct value in the target byte and random values in the remaining seven bytes were attempted, but they did not produce good results, that is no peaks stood out. Therefore, probably, the peak shown in the chart of Fig. 4a is not linked to a single check of the target byte, but it is related to a sequential verification of the MAC. Differently, the situation appears clearer for the behaviour linked to the last byte of the MAC. When the correct value is used, the MAC check is passed and the received message is decrypted, then verifying the returned r_{ICC} (note that in the BAC attempts of our timing analysis this check basically fails, as in the preparation of D_{IFD} we have fixed a vector of 0x00 bytes for r_{ICC}, which will be essentially always different from the r_{ICC} sent by the chip at the beginning of the BAC attempt). So the outstanding peak is due to the extra decryption operation performed by the chip. Note that for this reason, a lower number of iterations is needed to make the peak stands out and in principle even one iteration could be enough.

We also have to point out a specific mechanism adopted by the examined Italian ePassports featured by the side channel. They counted the number of


Set D_{IFD}=[0xDD$_0$,0xDD$_0$,...,0xDD$_{31}$]
Prepare M_{IFD}[8]=[0xRR,0xRR,0xRR,...,0xRR]
Run ByteStatistics(D_{IFD},M_{IFD},0)
ICC
Identify the correct value (0xMM$_0$) for M_{IFD}[0].
Prepare M_{IFD}[8]=[0xMM$_0$,0xRR,0xRR,...,0xRR]
Run ByteStatistics(D_{IFD},M_{IFD},1)
ICC
Identify the correct value (0xMM$_1$) for M_{IFD}[1].
Prepare M_{IFD}[8]=[0xMM$_0$,0xMM$_1$,0xRR,...,0xRR]
Run ByteStatistics(D_{IFD},M_{IFD},7)
ICC
Identify the correct value (0xMM$_7$) for M_{IFD}[7].
M_{IFD}=[0xMM$_0$,...,0xMM$_7$]
M_{IFD} is a valid MAC of D_{IFD} for the ICC
(0xRR denotes a random byte)
Precompute M_{IFD}=MAC_{Kmac}(D_{IFD}) for all possible MRZ
M_{IFD}
D_{IFD},M_{IFD}
Cracking Machine
MRZ
of the ICC

M_{IFD}	MRZ
0xCF,0x3C,...,0x71	0123EP<<<0
0x6D,0xFE,...,0x5C	4567IT<<<1
⋮	⋮

Fig. 5. Possible attack against the Basic Access Control exploiting the identified side channel.

consecutive unsuccessful BAC attempts, for instance due to MAC check failures, and when a specific threshold was reached the chip response was heavily delayed, basically preventing a correct time responses collection for our analysis. To overcome this issue, we periodically ran a successful BAC protocol within our analysis to reset the failure counter. We will discuss later how this feature represents a valid countermeasure against possible attacks that try to exploit the side channel just presented. We also report that for a given ePassport, at the beginning of our timing analysis over the MAC bytes, we recorded slightly higher response times during the first iterations, but this did not affect the validity of our analysis.

4 Possible Attack

An attack could be mounted against those ePassports that are featured by the highlighted side channel to obtain their MRZ. We present it in Fig. 5. In the attack scenario, for a given ePassport, differently from the timing analysis presented in the previous section, the relative MRZ is not known, so it is not possible to prepare a priori a MAC with some correct bytes. The idea is to use our timing analysis to retrieve byte by byte the valid MAC of a specific message. First, a generic 32-byte vector D_{IFD} is set. Then our timing analysis is launched, giving as input D_{IFD} and a random MAC of 8 bytes, selecting the first byte of the

MAC as the target one. The resulting chart will highlight the right value for the first byte of the correct MAC of D_{IFD}. The process is repeated preparing a MAC with the identified correct value for the first byte and setting the second byte as target, getting the right value for it from the timing analysis. Iterating such process over all MAC bytes as shown in Fig. 5 the full correct MAC of D_{IFD} for the given ePassport is obtained.

Such information could then be exploited in two different ways. The pair (D_{IFD},M_{IFD}) could be given to a cracking machine where all possible MRZs are used to compute the MAC of D_{IFD} until the match with M_{IFD} is found [6]. Alternatively, since D_{IFD} can be fixed a priori, all its possible MACs are precomputed using all possible MRZs, then exploiting the stored data as lookup table. Note that both approaches are feasible only if the attacked ePassport is featured by a low entropy MRZ, that is the full set of strings representing all possible MRZs is exhaustively manageable, but as reported in Sect. 1 this is the case for ePassports issued by different countries. For instance, it has been reported that the MRZ entropy for ePassports issued by some countries can be around 40 bits [4,7]. In such a case, considering that a cracking machine could be able to test $\approx 2^{28}$ BAC keys per second [6], ≈1 h would be required for getting the MRZ, while a lookup table would require some TBs of precomputed data.

We remark that this attack performs better than a brute force approach based on BAC attempts against the victim ePassport in terms of number of queries sent to the chip. Indeed, assuming 5000 iterations in the timing analysis for each of the first seven bytes of the MAC and 500 iterations for the last byte, ≈9 million queries would be needed in total (we remind that in our analysis, for each MAC byte, each possible byte value between 0x00 and 0xFF is tested N times, where N = 'number of iterations'), that is by far less than the number of possible MRZs. It also has to be noted that, according to the times experienced with our set-up, a timing analysis of 5000 iterations on a MAC byte requires an interaction of ≈85 h with the attacked chip, so some days would be required for a complete attack as the one estimated above, which could be not easy to achieve in practice even if we try to depict a couple of attack scenarios in the next section. In addition, for our statistical analysis to be successful and usable for the designed attack, the chip response times should not be affected by artificial behaviours as it was for our examined ePassports featured by the side channel. Indeed, for them, after roughly 250 failed BAC attempts the chip responses were artificially delayed by some seconds, de facto preventing any successful statistical analysis.

5 Discussion

Our experiments have been conducted by a lab set-up, with the examined chips interrogated through a contactless reader connected to a workstation. As real life attacking scenarios, we could think of NFC-enabled mobile phones used to mount attacks against BAC-protected chips. For instance two attack methodologies could be adopted, one based on physical proximity and the other acting remotely. In the first case, a person regularly in proximity of the victim document holder (e.g., on means of public transport, at the workplace) could collect data day by day. If the holder keeps for instance his document in a bag or in a pocket, the attacker could silently put his phone close to such points running the timing analysis in chunks, retrieving in the end the MRZ of the victim's document. For the second methodology an attacker should be able to install the timing analysis software in the mobile phone of the document holder. This could be achieved in different ways, as for instance distributing the analysis software through phone application repositories (e.g., hiding the software in games) or using social engineering techniques. Then the timing analysis is carried on when the victim keeps his phone close to the document (e.g., many people tend to keep phone and wallet, where the document could be placed, close together). Note that with such an approach it is possible to think of massive attacks infecting a large number of phones. Once cracked the document could be partially copied reading its content or remotely used through a relay attack [10].

We have got our timing analysis results on documents currently in circulation. We hope that our results foster ePassport chip manufacturers to assess their implementations in order to identify the possible presence of the side channel presented here. The issue could be solved forcing the MAC check to be executed in constant time or simply adding a delay for the chip responses when a certain number of unsuccessful BAC attempts have been run, de facto preventing the timing analysis. We have verified that this second option was adopted by the ePassports examined during our experiments, basically protecting them from our attack (even if such mechanism was probably introduced to protect the chip against brute force attacks). Also the adoption of high entropy MRZs prevents the attack, but a change in the MRZ scheme should be decided by the administrations of the different countries. In addition we remark as for ePassports the BAC is going to be replaced by the PACE scheme [11] that relies on a different cryptographic protocol.

Another electronic document that could be affected by our results is the electronic Driving Licence (eDL), recently regulated in the EU [12]. Similarly to ePassports a chip can be embedded in the document. Whether a contactless chip is adopted (also contact chips are possible) the access to its data is protected by the Basic Access Protection (BAP) [13]. Also the BAP is based on a shared secret between chip and reader, called Scanning Area Identifier (SAI), to run an authentication protocol between the parties. The BAP can be configured to act exactly as the BAC, the BAP 1 configuration of [13] specifies the same cryptographic algorithms, and also the SAI can be set to be a machine readable string. Such arrangement is exactly the one adopted for European eDLs, in favor of interoperability with existing equipment already used to read similar documents like ePassports, and no alternative options are given. In light of this a check on the SAI entropy and an assessment of the eDL chip implementation would be advisable, in order to evaluate the feasibility of the attack presented here against eDLs.

6 Conclusion

In the paper we present a side channel analysis for electronic documents featured by a contactless chip protected through the Basic Access Control (BAC). In particular specific timing analysis during chip operations for the BAC can be carried out. We explain how such analysis could be exploited to mount an attack to retrieve the chip's BAC keys when no countermeasures are adopted in combination with low entropy secrets. We have verified the presence of such side channel in ePassport chips currently in circulation and we remind that the same access control mechanism is adopted for contactless electronic Driving Licences. We advice all those players in charge of manufacturing and issuing such electronic documents to assess their security in light of these new results.

Acknowledgments. We thank Philippe Teuwen for his suggestions about the use of libnfc.

References

1. International Civil Aviation Organization: Machine Readable Travel Documents. Part 1, vol. 1, Sixth Edition (2006)
2. International Civil Aviation Organization: Machine Readable Travel Documents. Part 1, vol. 2, Sixth Edition (2006)
3. Juels, A., Molnar, D., Wagner, D.: Security and privacy issues in e-Passports. In: Proceedings of the IEEE 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks, pp. 74–88 (2005)
4. Avoine, G., Kalach, K., Quisquater, J.-J.: ePassport: Securing international contacts with contactless chips. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 141–155. Springer, Heidelberg (2008)
5. Hoepman, J.-H., Hubbers, E., Jacobs, B., Oostdijk, M., Schreur, R.W.: Crossing borders: Security and privacy issues of the European e-Passport. In: Yoshiura, H., Sakurai, K., Rannenberg, K., Murayama, Y., Kawamura, S. (eds.) IWSEC 2006. LNCS, vol. 4266, pp. 152–167. Springer, Heidelberg (2006)
6. Liu, Y., Kasper, T., Lemke-Rust, K., Paar, C.: E-Passport: Cracking basic access control keys. In: Meersman, R. (ed.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1531–1547. Springer, Heidelberg (2007)
7. Sportiello, L.: Weakening ePassports through bad implementations. In: Hoepman, J.-H., Verbauwhede, I. (eds.) RFIDSec 2012. LNCS, vol. 7739, pp. 123–136. Springer, Heidelberg (2013)
8. Chothia, T., Smirnov, V.: A traceability attack against e-Passports. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 20–34. Springer, Heidelberg (2010)
9. libnfc: Public platform independent Near Field Communication (NFC) library, Version 1.7.0 (2014). http://nfc-tools.org/
10. Sportiello, L., Ciardulli, A.: Long distance relay attack. In: Hutter, M., Schmidt, J.-M. (eds.) RFIDsec 2013. LNCS, vol. 8262, pp. 69–85. Springer, Heidelberg (2013)
11. International Civil Aviation Organization: Supplemental Access Control for Machine Readable Travel Documents, version 1.01 (2010)

12. Commission Regulation (EU) No. 383/2012: Laying down technical requirements with regard to driving licences which include a storage medium (microchip), 4 May 2012
13. ISO/IEC 18013: Information Technology - Personal Identification - ISO-Compliant Driving Licence - Part 3: Access Control, Authentication and Integrity Validation (2009)