A Survey of Security and Privacy Issues in ePassport Protocols

GILDAS AVOINE, INSA Rennes, IRISA UMR 6074, Institut Universitaire de France
ANTONIN BEAUJEANT, Université catholique de Louvain, Louvain-La-Neuve, Belgium
JULIO HERNANDEZ-CASTRO and LOUIS DEMAY, School of Computing, University of Kent, Canterbury, UK
PHILIPPE TEUWEN, NXP Semiconductors, Leuven, Belgium

This article examines in great detail the most relevant security and privacy issues affecting the protocols used by contactless chips integrated in ePassports, and presents all relevant literature together with some new attacks and insights that could help in improving future standards and the next generations of ePassports.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; E.3 [**Data Encryption**]

General Terms: Security, Design, Algorithms, Standardization, Experimentation

Additional Key Words and Phrases: Information security, cryptography, privacy, identification of persons, smartcards, forensics

ACM Reference Format:
Gildas Avoine, Antonin Beaujeant, Julio Hernandez-Castro, Louis Demay, and Philippe Teuwen. 2016. A survey of security and privacy issues in ePassport protocols. ACM Comput. Surv. 48, 3, Article 47 (February 2016), 37 pages.
DOI: http://dx.doi.org/10.1145/2825026

1. INTRODUCTION

Since King Artaxerxes I of Persia issued a letter to Nehemiah for traveling to Judea, sometime around 450 B.C., directed to "the governors beyond the river" [American Bible Society 1999], many things have changed through the world's history. What has not changed is the need for states and governments to offer some sort of certificate of identity and nationality to their citizens for easing international travel. There are numerous accounts of passports being used in Medieval times, from the Islamic Caliphate to Medieval Europe, generally in the form of Safe Conducts [Parliament of England 1414]. This consisted generally of a list of cities that the document owner was allowed to visit, or notes signed by the monarch or other local rulers. In Britain, for example, passports were first mentioned in the Safe Conduct Act of 1414. Only 500 years later, in 1914 and mostly due to World War I, they started including a photograph of the owner. For obvious reasons, passports were written using a number of languages. In the case of the British passport, it included Latin until 1772; then, French was used up

Authors' addresses: G. Avoine, IRISA Rennes, Campus universitaire de Beaulieu, 263 Avenue du Général Leclerc, CS 74205, Building F, Office 421 (red floor), 35042 Rennes Cedex, France; email: gildas.avoine@irisa.fr; A. Beaujeant, Université catholique de Louvain, Information Security Group, Place Saint Barbe, 2 Building Réaumur bte L5.02.01, B-1348 Louvain-la-Neuve, Belgium; email: antonin.beaujeant@gmail.com; J. Hernandez-Castro and L. Demay, School of Computing, University of Kent, CT2 7NF Canterbury, UK; emails: {jch27, ljfd2}@kent.ac.uk; P. Teuwen, Quarkslab, 71 Avenue des Ternes - 75017 Paris, France; email: pteuwen@quarkslab.com.

DOI: http://dx.doi.org/10.1145/2825026

to 1855. Security has always been one of the paramount issues surrounding passports, with the steady introduction of more sophisticated measures to make forgeries harder. One particularly extreme case is that of the Nicaraguan passport, which boasts 89 independent security features [Benedictus 2006] and was reputed to be one of the least forgeable documents in the world.

It is a common misconception that the electronic passport was introduced due to the September 11, 2001 (9/11) terrorist attacks. Davida and Desmedt [1988] suggested using a chip in passports as early as 1988. Malaysia issued electronic passports as early as 1998, and the International Civil Aviation Organization (ICAO) started working on standards late in the 1990s as well, although the first ICAO compliant passport (ePassport) was not issued until 2004, in Belgium. What is certainly true is that 9/11 and other terrorist attacks (Madrid 2004 and London 2005) sped up its deployment and extensive adoption, supported by their inclusion in previously existing initiatives such as the United States visa waiver program[1].

We will see in the rest of this article that the lack of a common application of the standard, which leaves too much room for different implementations, together with some design mistakes, make these new documents way less secure than they should be. We will concentrate on the technical details of the security solutions proposed, but we also want to highlight here that the overall security of current ePassport systems will, in all likelihood, be much lower than the one implied by our attacks. Our analysis focuses mostly on the protocols used by the ePassport, while other classes of attacks using completely different and unrelated approaches and tools also exist. As an example, we will not consider vulnerabilities such as those exposed by the Government Accountability Office (the United States Congress's investigative arm) in 2009 that, for example, discovered serious systemic problems such as the possibility of using fake documents (social security numbers from dead people, fake birth certificates or drivers' licenses) to get a genuine identification card, which then can be used to apply for an ePassport and successfully pass the enrollment phase [Sullivan 2009]. Also, we will not consider attacks performed during the personalization process, side-channel attacks on chips (see, e.g., Oswald et al. [2007]), denial-of-service attacks physically targeting the transponder or targeting the RF-interface (see, e.g., Finkenzeller [2009]), attacks on the inspection systems (see, e.g., the international interoperability tests that are regularly performed), and so on.

On the other hand, the scope of this article will not focus entirely on security issues. A related topic that is frequently overlooked, despite its increasing importance, is privacy. The vast majority of citizens seem to be less concerned about privacy than the Swiss: on May 17, 2009, Switzerland organized a vote on the introduction of the new generation ePassports (with fingerprints and EAC). It was a Pyrrhic victory, with 49.9% of the ballots cast against the introduction of the new ePassport[2]. Another recent case showing how privacy is routinely neglected is the introduction of more than 200 "smart" trash bins (Figure 1) in London, capable of tracking passersby by identifying their smartphone's Wi-Fi connections through its Media Access Control address (MAC address)[3] with the declared intention to "sell this information to brands to create targeted advertisements." Something similar could be done with ePassports, albeit generally only at a shorter range. Additionally, getting uniquely identifying information from a passport is nearly impossible without the knowledge of the MRZ.

Despite all concerns, these new documents equipped with a microchip and biometric data have been described in numerous standards and quickly adopted by an

[1]http://travel.state.gov/content/visas/english/visit/visa-waiver-program.html.
[2]Parlement Suisse. Votation populaire du 17 mai 2009. http://www.parlament.ch, May 2009.
[3]http://www.independent.co.uk/life-style/gadgets-and-tech/news/londons-bins-are-tracking-your-smart phone-8754924.html.

Fig. 1. A smart bin collecting passersby data in Central London. Credit: Renewlondon.com.

increasingly large number of countries. So much so that criminals are increasingly interested in obtaining them, as shown by the robbery [Topping 2008] of around 3,000 blank UK passports (apparently worth at least £2.5 million in the black market) destined for British embassies around the world.

The rest of the article is organized as follows: Section 2 offers a detailed exposition of the main ICAO specifications relevant to ePassports. Section 3 covers the basic ICAO security measures for ePassports, and Section 4 focuses in some depth on important additional security measures to further prevent illegitimate access to the passport's contents. Section 5 analyzes the multiple ISO standards that are relevant to the different components of ePassports. In Section 6, we present an exhaustive account of all publicly disclosed security vulnerabilities affecting the ePassport protocols. We analyze them in detail in Section 7, showing how to implement these and some completely new attacks based on these vulnerabilities. We also provide a comprehensive discussion regarding many anomalies, inconsistencies, and implementation pitfalls encountered when practically attacking passports in a lab setting. In Section 7, we also offer some implicit and explicit recommendations to improve the security of future ePassport standards, which are discussed more in-depth in Section 8. We present our conclusions in Section 9.

2. INTERNATIONAL CIVIL AVIATION ORGANIZATION

The ICAO, short for International Civil Aviation Organization, is the organization responsible for standards and recommended practices for air navigation, including those for the inspection of border crossing. The ICAO defined a set of standards for Machine Readable Travel Documents (MRTD) in Document 9303 and later also approved a standard for biometric passports. These, also known as electronic passports (ePassports), embed a Radio Frequency Identification (RFID) transponder in order to increase security and improve identification accuracy with biometric data (i.e., face (mandatory), fingerprints, and images of the iris).

2.1. Radio Frequency Identification

RFID is a contactless identification technology that uses an electromagnetic field to communicate between the identification system and the data carrier (i.e., transponder or tag). There are two categories of transponders: active and passive ones. Active tags carry a battery and allow for long-range communication; passive tags, such as the ones used in ePassports, harvest energy from the interrogation zone[4] to power the chip on and establish a communication with the reader. Consequently, passive tag technology works only at a relatively short range from the reader. Communication and technical standards have been chosen in accordance with the requirements of the biometric

[4]The interrogation zone is the operational zone where the electromagnetic field generated by the reader is strong enough to establish a communication between both parties.

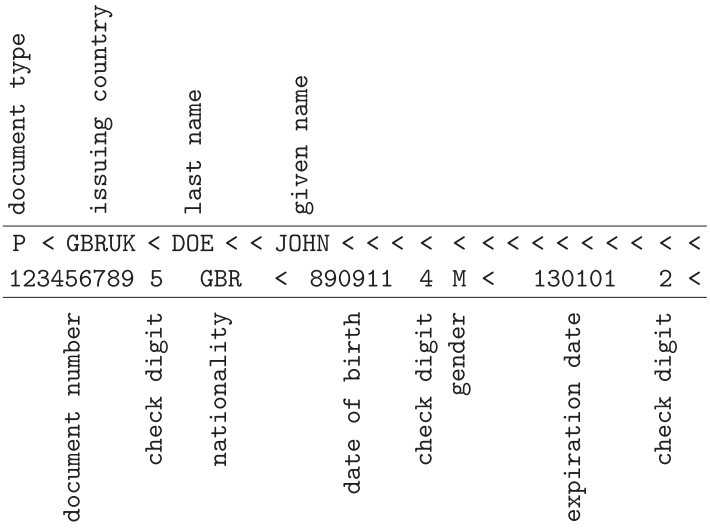

Fig. 2. MRZ description.

passport, such as the need for cryptographic algorithms (e.g., RSA and 3DES), the content or the position (usually in the back cover) of the transponder, and so on.

The ICAO selected RFID as the ideal technology to be used in the digital communication between the inspection system and the biometric passport due to its many advantages: it is contactless, has no need to have a line of sight for reading, has no need for a battery, is capable of executing heavy cryptographic primitives, and is relatively cheap.

Even though this technology extends the security[5] of ePassports, it exposes them to a new and frequently not well understood range of attacks and privacy issues, mostly derived from its contactless features.

2.2. Document 9303

ISO/IEC 7501-1, also known as Document 9303, is a standard developed by the ICAO that describes MRTDs. Part 3, *Machine Readable Official Travel Documents*, is split into 2 volumes: ICAO [2008a] and ICAO [2008b]. Volume 1 describes the *Machine Readable Zone* (MRZ): a zone with two *Optical Character Recognition* (OCR) lines of 44 characters each, used by the inspection system to read the passport. As represented in Figure 2, the first line is composed of the document code and the name, and the second provides personal information about the bearer, that is document number (and check digit), nationality, date of birth (and check digit), sex, passport expiration date (and check digit), personal number (and check digit) and the overall checksum. Each character is either alphabetical (A-Z), numerical (0-9), or the filler character (<).

Biometric passports embed data in order to identify its owner. As illustrated in Table I, the data is structured in different Data Groups (DGs) indexed in a presence map named *Common*. Each DG present in the passport has its hash in a Document Security Object (SO_D) in order to verify its integrity. Two DGs are mandatory: DG1, which gathers all the information printed on the passport, and DG2, which contains the owner's encoded face (in a JPEG or JPEG2000 file).

[5]Before the appearance of the biometric passport, security relied solely on physical anticounterfeiting techniques for authentication, and face comparison for identification.

Table I. Logical Data Structure (LDS): Content and File IDentifier (FID) of DGs

Data Group (DG)	FID	Content
Common	'01 1E'	LDS (Logical Data Structure) Version number
		Unicode Version number
		List of all Data Groups present
DG1	'01 01'	MRZ
		Document Type
		Issuing State / organization
		Name (of Holder)
		Document Number
		Check Digit - Doc Number
		Nationality
		Date of Birth
		Check Digit - DOB (Date of birth)
		Sex
		Data of Expiry
		Check Digit - DOE (Date of Expiry)
		Optional Data
		Composite Check Digit
		MRZ contents specification
DG2	01 02	Encoded Face
DG3	01 03	Encoded Finger(s)
DG4	01 04	Encoded Eye(s)
DG5	01 05	Displayed Portrait
DG6	01 06	Reserved for Future Use
DG7	01 07	Displayed Signature or Usual Mark
DG8	01 08	Data Feature(s)
DG9	01 09	Structure Feature(s)
DG10	01 0A	Substance Feature(s)
DG11	01 0B	Additional Personal Detail(s)
DG12	01 0C	Additional Document Detail(s)
DG13	01 0D	Optional Detail(s)
DG14	01 0E	Security Options for Secondary Biometrics
DG15	01 0F	Active Authentication Public Key Info
DG16	01 1D	Person(s) to Notify
Security Object	01 1E	Carries the hashed LDS Data Groups

3. SECURITY

Four security mechanisms are defined by the ICAO [ICAO 2008b]: *Basic Access Control* (BAC), *Passive Authentication* (PA), *Active Authentication* (AA) and *Extended Access Control* (EAC). The ICAO requires only the PA (although the other security mechanisms are strongly recommended), while the EU [European Commission 2009] requires member states to also implement the BAC.

3.1. Basic Access Control

The BAC depicted in Figure 4 aims to ensure that reading the content of the chip (an operation performed typically by officers at the border) is possible only if the document is open onto the data page. This tries to prevent unauthorized distant readings. During the BAC process, only the second line of the MRZ is read, which is then used to derive two cryptographic keys needed for accessing the content of the chip.

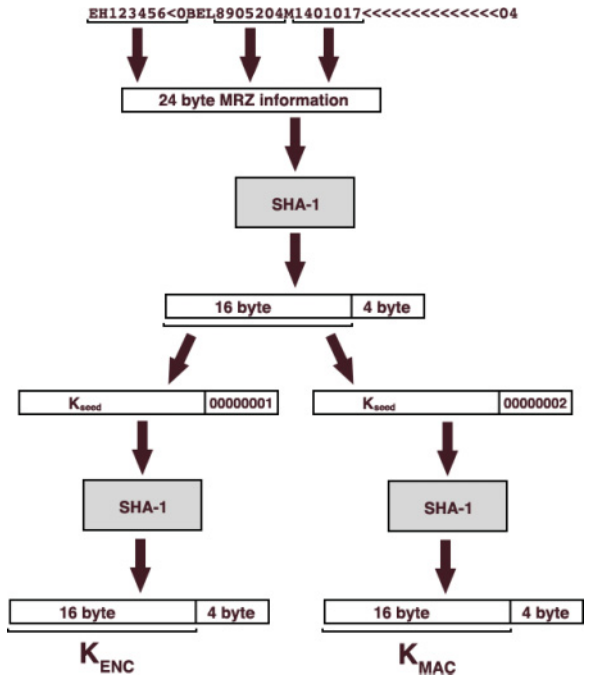

Fig. 3. K_{ENC} and K_{MAC} derivation.

3.1.1. Key Derivation. As illustrated in Figure 3, the inspection system hashes the MRZ_information[6] with SHA-1. The 16 most significant bytes form K_{seed}, which is then concatenated to 00000001 ($K_{\mathrm{seed}} \| 00000001 = K_{\mathrm{a}}$) and 00000002 ($K_{\mathrm{seed}} \| 00000002 = K_{\mathrm{b}}$). The SHA-1 hash of K_{a} is computed, and the 16 most significant bytes form the 3DES encryption key K_{ENC}. The same process is performed with K_{b} to form the MAC key K_{MAC}.

3.1.2. Protocol. The passport generates an 8B random number ($\mathrm{RND}_{\mathrm{ICC}}$) and sends it to the inspection system (IS). In practice, this challenge is the passport answer after receiving the GET_CHALLENGE command [ISO 2005].

Upon reception of $\mathrm{RND}_{\mathrm{ICC}}$, the IS generates an 8B random number ($\mathrm{RND}_{\mathrm{IFD}}$) and a 16B key ($K_{\mathrm{IFD}}$), concatenates these values ($\mathrm{RND}_{\mathrm{IFD}} \| \mathrm{RND}_{\mathrm{ICC}} \| K_{\mathrm{IFD}}$), encrypts the result using 3DES with K_{ENC}, and computes the 8B MAC with K_{MAC}. The computer ciphertext and MAC are sent to the passport. The passport computes the MAC of the received ciphertext with K_{MAC} and checks if it matches the MAC received. If so, the passport decrypts the ciphertext with K_{ENC} and checks whether $\mathrm{RND}'_{\mathrm{ICC}}$[7] matches the random number generated previously.

The passport then generates a 16B key K_{ICC} and concatenates it with the two random numbers ($\mathrm{RND}_{\mathrm{ICC}} \| \mathrm{RND}'_{\mathrm{IFD}} \| K_{\mathrm{ICC}}$). The obtained value is encrypted using 3DES with K_{ENC} and MACed with K_{MAC}. The ciphertext and the MAC are concatenated and sent to the IS.

The IS finally verifies the MAC, then decrypts the message and checks whether both random numbers match those stored from the previous steps. The K_{ICC} is XORed

[6]MRZ_information is the concatenation of the document number + check digit with the date of birth + check digit and the expiration date + check digit.
[7]The received values are noted with a prime.

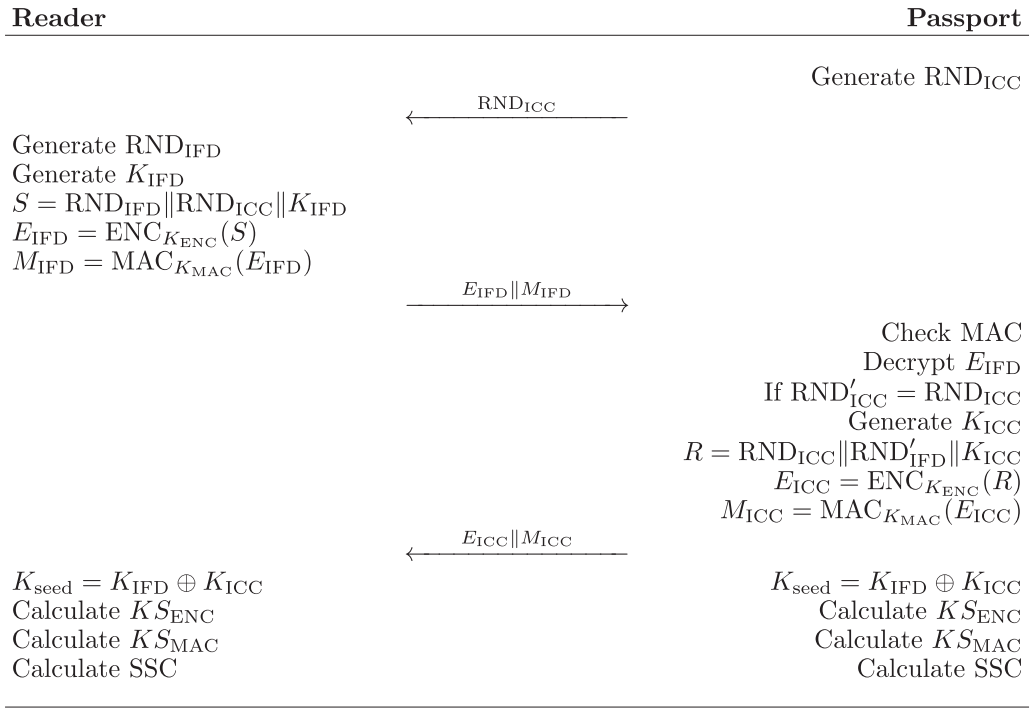

Fig. 4. Basic Access Control.

with K_{IFD} to form the key seed used for the derivation of the session keys KS_{ENC} and KS_{MAC}. The derivation process is the same as for K_{ENC} and K_{MAC}. The final step is the calculation of *Send Sequence Counter* (SSC): the concatenation of the 4 less significant bytes of $\mathrm{RND}_{\mathrm{ICC}}$ and the 4 less significant bytes of $\mathrm{RND}_{\mathrm{IFD}}$.

3.1.3. Secure Messaging. Once the session keys and the SSC are computed, the communication is protected by the secure messaging mechanism, for which KS_{ENC} is the encryption key and KS_{MAC} the integrity key: the encryption uses the 3DES algorithm in CBC mode with a zero IV (i.e., `0x00 00 00 00 00 00 00 00`), and a padding compliant with ISO/IEC 9797-1 padding method 2. The MAC is computed using ISO/IEC 9797-1 MAC algorithm 3 with the DES block cipher, a zero IV, and the ISO/IEC 9797-1 padding method 2 [ICAO 2008b]. Before the MAC is computed, the SSC is incremented and prepended (attached as a prefix) to the ciphertext.

3.2. Passive Authentication

Passive authentication is a mandatory security mechanism whose aim is to ensure the authenticity and integrity of the passport contents. The authentication consists of verifying a signature value stored in the passport, computed by the issuing country with a key common to lots of passports issued during the same period of time. Passive authentication aims to stop a counterfeiter from modifying the contents of a passport or from creating a new passport on one's own, but it does not stop a miscreant from cloning a genuine passport.

When a passport is personalized, each DG is generated and cryptographically hashed. The hash values are encapsulated in a signed data structure compliant with RFC 3369, and stored in the *Elementary File Document Security Object* (EF.SOD) of the passport.

The data structure is then signed, and the signature is included into the EF.SOD. The signature can be checked by the inspection system using the *Document Signer* (DS) X.509 certificate, also available from the EF.SOD along with the hash values and signature. Alternatively, the DS certificates can be obtained from the ICAO *Public Key Directory* (PKD). In turn, a DS certificate can be checked using the *Country Signing Certificate Authority* (CSCA) X.509 certificate.

According to Document 9303, passive authentication should be based on one of the following signature schemes: RSA, DSA, or ECDSA. States implementing RSA should use RFC 3447 PKCS#1. Document 9303 recommends using RSASSA-PSS with a modulus larger than 3 072b, and 2 048b for the CSCA and the DS keys. States implementing DSA should use FIPS 186-2. The minimum sizes for the moduli p and q should be 3 072b and 256b, and 2 048b and 224b for the CSCA keys and DS keys, respectively. Finally, states implementing ECDSA should use X9.62 or ISO/IEC 15946. The minimum size for the base point order should be 256b and 224b for the CSCA keys and the DS keys, respectively. The hash algorithm should be selected among the following, in accordance with the selected signature scheme: SHA-1, SHA-224, SHA-256, SHA-384, or SHA-512 (FIPS 180-2).

Document 9303 also provides guidelines for key management: it recommends renewing the CSCA keys every three to five years, and the DS keys every three months. Also, the ICAO PKD should not publish the CSCA certificates, but rather use them for checking the received DS certificates before publication. The CSCA certificates should, in accordance with bilateral agreements, be exchanged between states. This recommendation made the spreading of the CSCA certificates quite inefficient. This is a very relevant issue because the security introduced by ePassport usage is effectively null and void without these root certificates. Fortunately, some of these recommendations were disobeyed relatively soon, as several countries decided to publicly release their own CSCA certificates.

Following the need to improve the diffusion of CSCA certificates, ICAO enforced the prerogatives of the ICAO PKD based on an initiative of a few countries (Australia, Canada, New Zealand, United Kingdom, United States, and Singapore). Today, the ICAO PKD contains DS certificates, *Certification Revocation Lists* (CRLs), CSCA Link certificates (a means to replace a former CSCA certificate by a new one, by simply signing the new CSCA certificate with the former one), and *Master Lists* of CSCA certificates. A Master List is a list of certificates signed by a trusted country. The Master List is in some way related to the well-known concept of web-of-trust in OpenPGP: a party A can use the certificate of an unknown party B if this certificate is signed by a peer C trusted by A.

3.3. Active Authentication

The AA depicted in Figure 5 aims at verifying that the passport has not been cloned, thanks to a secured memory that contains a private key unique to each passport and a public key located in DG15.

3.3.1. Protocol. In order to authenticate the passport, the reader generates an 8B random number M_2 that is sent using the *INTERNAL AUTHENTICATE* command [ISO 2005].

Upon reception of M_2', the passport creates a trailer T including the hash algorithm used for the AA (e.g., the trailer is `0xBC` for the hash algorithm SHA-1) and generates a random number M_1 of length L_{M1}. The passport hashes the concatenation of M_1 and M_2'. Then, it constructs the message representative, which is the concatenation of the header (`0x6A`), M_1, the hash (H), and the trailer. The length L_{M1} is such that this message does not need padding (e.g., L_{M1} is calculated such that the length of the message is exactly 1024b when RSA-1024 is used).

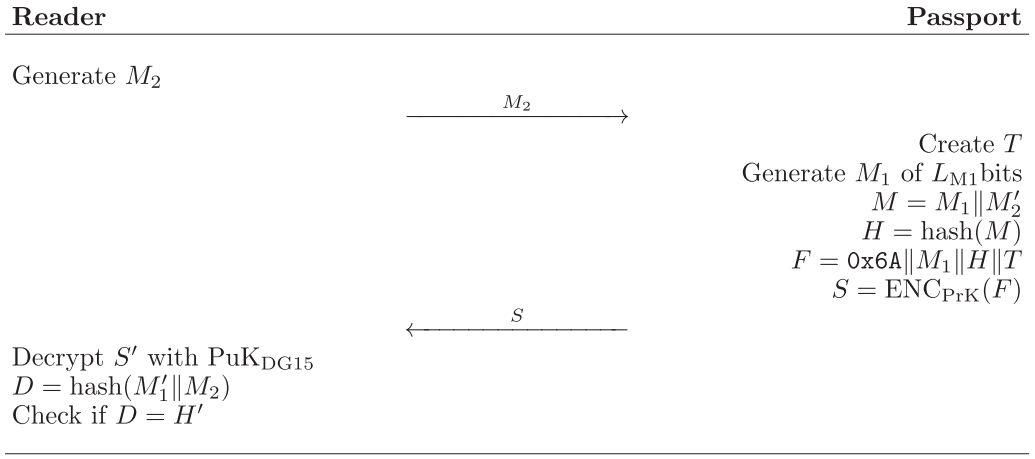

Fig. 5. Active Authentication.

Finally, the passport signs the message representative with the protected AA private key, and sends the signature to the reader. The reader verifies the signature with the public key stored in DG15. Depending on the value of the trailer, the reader selects the hash algorithm to use, extracts the digest (H') and M_1', then it hashes the concatenation of M_1' and M_2 to get a new digest (D). If D matches H', the tag is authenticated.

The AA requires checking the PA as well, otherwise this mechanism would be insecure. Indeed, the signature provided by the passport during the AA must be checked using the passport's public key stored in DG15. Integrity and authenticity of this public key are consequently ensured by and only by the PA. Note that DG15 contains the public key only, without a certificate.

As the private key is not readable and should be impossible to infer from the computed messages, a cloned passport will fail at the AA phase.

4. ADDITIONAL SECURITY MEASURES

Biometric passports allow for the storage of quite sensitive data, such as biometric data (i.e., fingerprints and images of the iris) that should not be available to everyone, not even to people who may have a line of sight inside the passport (e.g., acquaintances or thieves). In order to prevent unwanted read access to this sensitive information, the ICAO recommends additional security measures. There are three generations of security solutions in ePassports. BAC, introduced in 2005, was later extended in 2009 to EACv1 and EACv2. SAC was launched in 2014, and uses PACE as an additional security mechanism, aiming to provide secure ePassports for the next 20 years or so. SAC improves BAC with more entropy. EAC adds terminal authentication (TA), and provides a stronger session key than BAC. Additionally, the CA part of EAC does not bring the challenge semantics problem (described in BSI [2015]) that has been identified on AA.

4.1. Supplemental Access Control and Password Authenticated Connection Establishment

Password Authenticated Connection Establishment (PACE, depicted in Figure 6) is a countermeasure to the weakness of the BAC, in which strong session keys are computed despite the low entropy of the shared secret (e.g., 6 digits are sufficient in general). As for BAC, the shared secret is derived from the MRZ information, but it can also optionally be derived from the Card Access Number (CAN). The CAN is printed in the

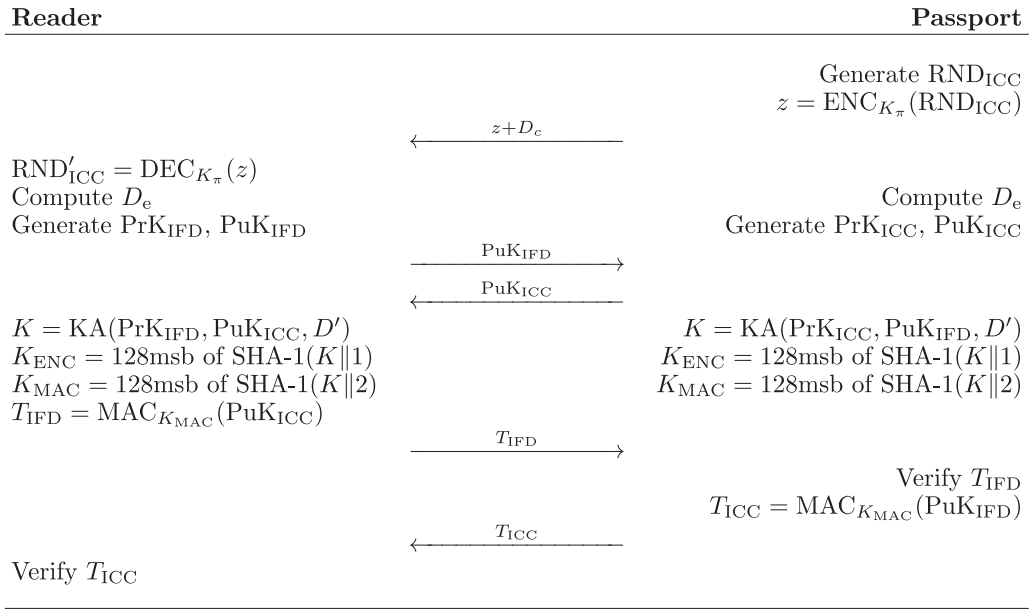

Fig. 6. Password Authenticated Connection establishment.

passport and can be read easily, avoiding the inspection system for using an optical MRZ reader, while such a reader is required to obtain the MRZ.

In 2010, ICAO introduced Supplemental Access Control (SAC) [ICAO 2010], based on PACEv2. SAC must be implemented by EU countries by the end of 2014. According to R12-TR_SAC_0006 in ICAO [2013b], by the end of 2017, only ePassports supporting SAC will be considered to be ICAO compliant. Meanwhile, ePassports will still have to support BAC to allow time for all inspection systems to change to SAC-only passports.

4.1.1. Protocol. As for the BAC, the shared secret (π) is printed on the passport. The inspection system reads the password (i.e., CAN or MRZ-Password) π with an OCR scanner (or manually) and computes the key $K_\pi = \text{SHA-1}(\pi\|3)$. The passport generates a random number ($\mathrm{RND_{ICC}}$), encrypts it with K_π and sends it to the reader together with the domain parameters (D_c). The reader decrypts the received value with K_π to get $\mathrm{RND_{ICC}}$. Both the reader and the passport compute the ephemeral domain parameters D_e based on $\mathrm{RND_{ICC}}$ and D_c, then generate a pair of keys: $\mathrm{PrK_{ICC}}$, $\mathrm{PuK_{ICC}}$ and $\mathrm{PrK_{IFD}}$, and $\mathrm{PuK_{IFD}}$, and share their public keys. A common key K is established using a key exchange (KA) (either by the standard Diffie-Hellman protocol or an elliptic curve key exchange protocol [Belguechi et al. 2012]) such as:

$$K = \mathrm{KA}(\mathrm{PrK_{ICC}}, \mathrm{PuK_{IFD}}, D') = \mathrm{KA}(\mathrm{PrK_{IFD}}, \mathrm{PuK_{ICC}}, D')$$

The session keys are then computed: K_ENC and K_MAC consist of the 16 most significant bytes of, respectively, SHA-1$(K\|1)$ and SHA-1$(K\|2)$. The reader computes an authentication token T_IFD, which is a MAC of the passport public key ($\mathrm{PuK_{ICC}}$) computed with the key K_ICC. The reader then sends the MAC value to the passport. Finally, the passport verifies T_IFD, then generates an authentication token T_ICC and sends it to the reader, which verifies T_ICC. After a successful PACE run, both parties share strong encryption and MAC keys to establish a secure communication channel.

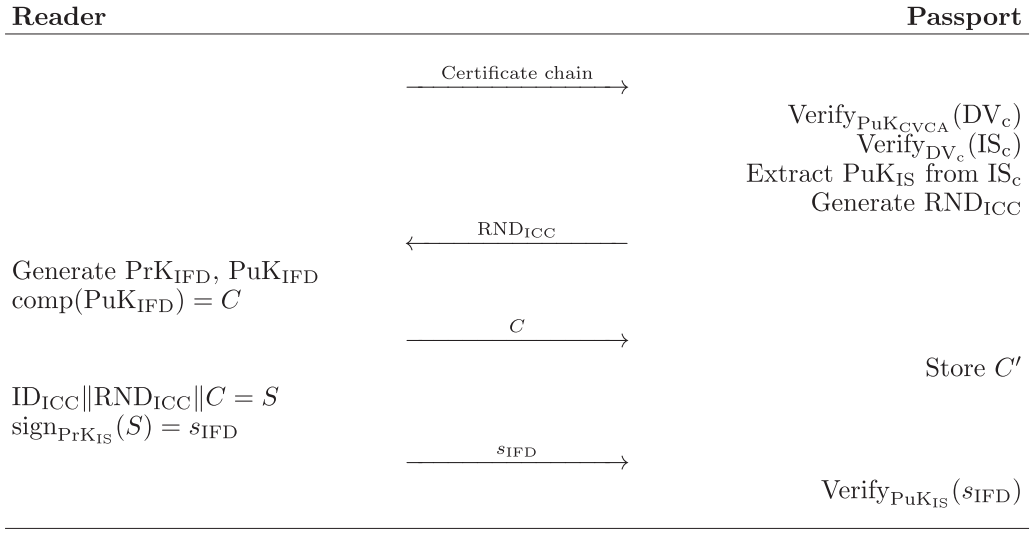

Fig. 7. Terminal Authentication, Version 2.

4.2. Extended Access Control

EAC uses a set of Document Extended Access Keys for encryption, instead of the Document Basic Access Keys (K_{ENC} and K_{MAC}) used in BAC. This set of keys may consist of either symmetric keys or an asymmetric key pair [ICAO 2008b]. It is up to the document issuer to implement and define this security measure. While several countries have developed their own EAC implementations, the German Federal Office for Information Security (BSI) prepared the specification for EU passports and published it under their technical guideline BSI TR-03110. EACv1, as defined for EU, contains the *Chip Authentication* (CA) and *Terminal Authentication* (TA) mechanisms. CA also establishes a secure channel between the reader and the passport, but enables the inspection system to authenticate the passport. TA may only be used together with CA. It enables the passport to verify that the IS is allowed to access sensitive information such as biometric data. There also exists an EACv2, extending the authentication of the terminals to access the ICAO-mandatory data groups (DG1, DG2, SOD). As those must be readable by countries that are not implementing EAC, however, EACv2 is not used for ePassports, rather is used mostly for eID cards [Mösenbacher 2013]. Nevertheless, BAC is quickly being phased out and replaced by SAC to secure access to the mandatory data groups.

The version defined by BSI [2009] requires that TA (Figure 7) must be performed before CA (Figure 8). The passport is authenticated with the CA, and new session keys are used when successfully performed; the inspection system is authenticated with the TA using inspection system certificates granted by the Document Verifier (DV), which is generally delivered by the Country Verification Certificate Authority (CVCA).

4.2.1. Terminal Authentication. Regarding the description of the BSI, each country that issues passports with EAC capabilities needs a CVCA: a one trust point that issues DV certificates. Upon creation, the public key of the CVCA is stored in the passport. A DV is the organizational unit that issues Inspection System (IS) certificates, used for the authentication of inspection systems: The reader (IS) initiates the terminal authentication by sending a certificate chain. The passport verifies the DV certificate with PuK_{CVCA} and the IS certificate with the DV certificate, then extracts the PuK_{IS} from

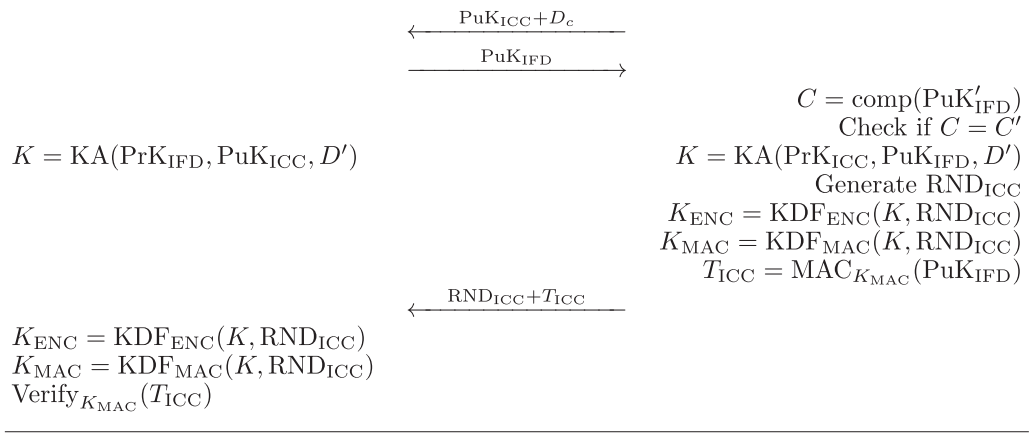

Fig. 8. Chip Authentication, Version 2.

the IS certificate. Finally, the passport generates a random number RND_{ICC} and sends it to the reader. Upon reception, the reader generates an ephemeral Diffie-Hellman key pair (PrK_{IFD} and PuK_{IFD}), hashes (compresses) its public key, then signs with PrK_{IS} the concatenation of the document number, including the check digit (ID_{ICC}), RND'_{ICC} and the digest. The signature is sent to the passport, which verifies it with PuK_{IS} as extracted previously from the IS certificate. Once the terminal authentication is successfully performed, the passport starts a CA in order to secure the communication.

4.2.2. Chip Authentication. Session keys K_{ENC} and K_{MAC} are derived from a shared secret K, which is computed via a Diffie-Hellman key agreement protocol (KA). The passport has static Diffie-Hellman public and private keys (PuK_{ICC} and PrK_{ICC}). It initiates the chip authentication by sending PuK_{ICC} together with the domain parameters D_c. The reader sends its ephemeral Diffie-Hellman public key (PuK_{IFD}), generated during the TA, to the passport. First, the passport hashes (compresses) PuK_{IFD} and checks if it matches the digest sent by the reader during the TA, then both the reader and the passport compute the shared secret K, as during the PACE. The passport generates a random number (RND_{ICC}), then derives the session keys using a key derivation function (KDF) such as:

$$K_{\text{ENC}} = \text{KDF}_{\text{ENC}}(K, \text{RND}_{\text{ICC}})$$

and

$$K_{\text{MAC}} = \text{KDF}_{\text{MAC}}(K, \text{RND}_{\text{ICC}}).$$

The passport generates an authentication token (as in the PACE process) then sends RND_{ICC} and the token (T_{ICC}) to the reader. The reader uses RND_{ICC} to derive the session keys, then verifies T_{ICC}. Finally, in order to verify the authenticity of the PuK_{ICC}, the reader performs a PA.

Belguechi et al. [2012] highlight an important issue about biometric data in identification systems: 'Since standard biometric templates are permanently associated with an individual, they could not be used any more in case they are compromised. Since they cannot be replaced, they are also inherently nonrevocable. This makes classical biometric systems inappropriate for privacy and security critical applications." A solution proposed in this article is to embed a cancellable biometric template of an individual in its electronic passport instead of the "raw" biometric data.

5. STANDARDS

5.1. ISO/IEC 14443

ISO/IEC 14443 defines the transponder and its transmission protocols. The standard is divided into four parts: (i) physical characteristics, (ii) radio frequency power and signal interface, (iii) initialization and anticollision, and (iv) transmission protocol. The standard describes two different tags: Type A and Type B, whose main differences are the modulation, coding, and protocol initialization procedures. It is up to the issuing country to decide which type to use for its passports.

ISO/IEC 14443 Part 3 covers the anticollision system [ISO 2011]. Whenever several transponders enter the interrogation zone, the reader has to identify each tag in order to communicate with one at a time. For that, each transponder uses a unique identifier (UID).

ISO/IEC 14443 Part 4 covers the transmission protocol [ISO 2008]. It defines how to transfer the Application Protocol Data Units (APDUs) as defined in ISO/IEC 7816-4 for the protocol $T = 1$. It also defines Type A *Answer to Select* (ATS) and Type B protocol info bytes.

5.2. ISO/IEC 7816

The transponder embedded in a biometric passport is compliant with ISO/IEC 14443-4, and therefore with ISO/IEC 7816-4. The ISO/IEC 7816 standard is organized into 15 parts but only Part 4 and, incidentally, Part 3 are relevant for our study.

5.2.1. ISO/IEC 7816-3 and PC/SC Part 3. ISO/IEC 7816-3 covers the electrical interface and transmission protocols [ISO 2006]. An ISO/IEC 7816 compliant transponder sends an *Answer to Reset* (ATR) upon an electrical reset (e.g., whenever the transponder enters the interrogation zone). The ATR provides information about the communication parameters and the transponder itself (e.g., version number). A passport does not comply with ISO/IEC 7816-3, however; thus, when using a reader is compliant with PC/SC, the reader interface handler has to generate a fake ATR for those ISO/IEC 14443 transponders. The forged ATR has a standard prefix concatenated with different data depending on the tag being a Type A or Type B. ATRs generated for ISO/IEC 14443 Type A tags embed the *historical data* from the ATS. ATRs generated for ISO/IEC 14443 Type B tags embed the *Application Data* and *Protocol Information* [PC/SC Workgroup 2007].

5.2.2. ISO/IEC 7816-4. The communication messaging protocol between a reader and a passport uses the APDU. There are two APDU categories: the command and the answer. A command APDU is sent by the inspection system, denoted interface device (IFD), to the passport, denoted integrated circuit card (ICC). The first 4B are the mandatory header (CLA, INS, P1, and P2). In the case of short length fields, the next 0B to 257B is the data. The first byte of the data is the payload length (L_C) and the last byte is the expected response length (L_E). The payload[8] is the 0B to 255B of long data between L_C and L_E. If the payload length is null, L_C is omitted. If the command does not expect data in the answer, L_E is omitted. A response APDU is sent by the ICC to the IFD. It contains 0B to 256B of data concatenated with a mandatory 2B status word (SW1 and SW2)[9].

A transponder may contain several applications; thus, before starting to use an application, the IFD has to select it through its Application Identifier (AID). The identifier

[8]Also known as *command data*.

[9]In the case of extended length fields, L_C can be 3B long and L_E can be 2B or 3B long depending on if L_C is present. Payloads can then be theoretically as large as 65535B.

number for the passport is composed of the Registered Application Identifier (RID) ‘A0 00 00 02 47’ and the Proprietary Application Identifier Extension (PIX) ‘1001’. The command for selecting the passport application is the *SELECT FILE* command:

CLA	INS	P1	P2	L_C	DATA	L_E
00	A4	04	0C	07	A0000002471001	–

There are three main steps to read an Elementary File: (1) select the file, (2) read the header, and (3) get the content. For the first step, the reader (IFD) sends the *SELECT FILE* command to the passport (ICC) with the data field set to the FID or the short EF identifier. If the EF exists, the passport sends a response APDU with the status *success* (0x90 0x00) and the data empty. The next step, that is, read the header, is required because a response command embeds a maximum of 256B or less (see FSD in ISO/IEC 14443 Part 4 [ISO 2008]). If a DG is longer, it will not fit in one APDU, thus the DG has to be retrieved through several APDUs. In order to know the file length, the IFD needs to read the header file that embeds its own length together with the file length. A header has a maximal length of 3B; therefore, the reader sends a *READ BINARY* command with L_E set to 0x03. Finally, for the last step, the reader will send as many *READ BINARY* commands as necessary to fetch the entire EF, where P1 and P2 specify an EF and/or an offset (see ISO/IEC 7816-4 ch7.2.2 [ISO 2005]). For instance, for the first command, the offset is set to the header size (1B, 2B, or 3B depending on the EF size).

6. VULNERABILITIES

Six vulnerabilities and their variants have been identified so far in the literature. These vulnerabilities were discovered either in the design of the security protocols or in their implementations. They are described in the coming sections according to their attack vectors. However, considering a classification based on the attack objectives might be useful as well. We thus provide here a list of attacks classified according to their objective.

—*Recognition of an individual passport:* traceability due to the AA (Section 6.5).
—*Recognition of a batch of passports (to infer the issuing country):* response-time vulnerability (Section 6.2) and fingerprinting methods (Section 6.3).
—*Obtaining a proof of presence:* signature evidence due to the AA (Section 6.5).
—*Obtaining the data contained in the passports:* low MRZ entropy (Section 6.1), MRZ brute forcing with a lookup table (Section 6.4)

6.1. MRZ Low Entropy

The BAC authenticates the reader as legitimate if it uses the proper K_{ENC} and K_{MAC} for the establishment of the session keys (i.e., KS_{ENC} and KS_{MAC}) used to read the passport contents. K_{ENC} and K_{MAC} are keys directly derived from the MRZ. This means that, if an attacker has read the MRZ, that attacker would have access to all DGs not protected with EAC. Only a few components from the MRZ are used during the generation of the session keys, however: (i) the document number, (ii) the date of birth, and (iii) the expiration date and their respective checksums. The name, nationality, sex, and other fields are irrelevant in this process. Checksums are redundant (they depend on aforementioned data), thus they do not increase the entropy at all. The three components left are highly structured and are far from being random:

The *document number* is at most 9 alphanumerical characters long. The maximal entropy for the document number is $(26 + 10)^9 \approx 1.02 \times 10^{14}$, or around 46.53b. Nevertheless, and depending on the issuing country, the document number can have a very

Table II. Command and Response APDU Description

Command APDU		
Field name	Length (byte)	Description
CLA	1	Instruction class (type of command)
INS	1	Instruction code (specific command)
P1-P2	2	Instruction parameters for the command
L_c	0, 1, or 3	Number of bytes of command data (N_c)
Command data	N_c	N_c bytes of data
L_e	0-3	Maximum number of response bytes expected (N_e)
Response APDU		
Response data	N_r (at most N_e)	Response data
SW1-SW2	2	Command processing status

specific structure. In Belgian passports, for instance, it is always made of 2 characters, followed by 6 digits. This significantly reduces the entropy to $26^2 \times 10^6 = 6.76 \times 10^8$, or around 29.33b. In French passports, it is composed of 2 digits, followed by 2 characters, followed by 5 digits, leading to an entropy of $26^2 \times 10^7 = 6.76 \times 10^9$, or around 32.65b. It is possible to reduce the entropy of the document number even more due to distribution patterns. Belgian passports seem to follow a sequential pattern distribution according to Avoine et al. [2008]. Indeed, passports with later issue dates are assigned higher document numbers. Based on this correlation, it would be possible, for instance, to establish a database that links document numbers with their issue or expiration dates. By sending the issuing country and the date of expiration of a passport, and after checking with said database, an attacker could easily obtain the structure and a range for the document number. The accuracy of the range depends on the number of entries in the database. A well-populated and up-to-date database might reduce the entropy to around 13b, very far from its maximal theoretical value of around 46.5b.

The date of birth is 6-digit long. It is a YYMMDD structured date. Only two digits are used to code the year; therefore, only 100 years are covered. There are 36525 days in 100 years. If the attacker knows the victim or how the victim looks like, the attacker might narrow down the range from 5 years ($5 \times 365 = 1825$ days) to 1 day.

The expiry date is structured similarly to the date of birth: in 6-digit YYMMDD format. Depending on the issuer country, the validity of passports is 5, 7, or 10 years. The entropy of the expiry date of a valid passport is $5 \times 365 = 1825$, or $10 \times 365 = 3650$, thus less than 12b. Research on the Belgium passport [Avoine et al. 2008] shows that they are never issued during weekends and holidays. This might be true for other countries as well. If so, this simple fact will reduce the number of potential issuing days in a year to about 250 days. As the date of expiration is always exactly 5, 7, or 10 years after the issuing date, the entropy of the expiry date is reduced to less than $5 \times 250 = 1250$, or $10 \times 250 = 2500$ possibilities, thus less than 11.28b.

The MRZ information used in the BAC for the derivation of K_{ENC} and K_{MAC} has a key space of 1.36×10^{22}, which can be reduced to 2.28×10^{10} under certain circumstances (see Table III). Systems with such a low key space are easy to bypass by means of an exhaustive key search, also known as a brute-force attack. This type of attack uses a predefined range, or a list, of keys and systematically checks all of them. Two families of practical attacks have been implemented in this way, solely based on this low-entropy vulnerability: the *online brute-force* attack, and the *offline brute-force* attack. A third attack type has been found by Sportiello [2012] using lookup tables, and is described in Section 6.4.

6.1.1. Online Brute-Force Attack. This attack uses a list of potential MRZ values. For each plausible MRZ, K_{ENC} and K_{MAC} are derived, and the BAC process is started.

Table III. MRZ_Information Key Space Summary

Fields	No information	Country known	Visual access
Document number	1.02×10^{14}	from 6.76×10^{8} to 10^{4}	–
Date of birth	36525 days	–	1825 days
Date of expiration	3652.5 days	from 3652.5 days to 1250 days	–
Total			
No information	Country known	Visual access	Country+Visual
1.36×10^{22}	from 9.02×10^{16} to 4.57×10^{11}	6.8×10^{20}	from 4.51×10^{15} to 2.28×10^{10}
73.52 bits	from 56.32b to 38.73b	69.20b	from 52.00b to 34.41b

The attacker then sends the first ciphertext, together with its MAC. Upon reception, the passport verifies the MAC: it generates the MAC based on the ciphertext with its K_{MAC}. If the MAC matches the one sent concatenated with the ciphertext, the BAC process continues and the key (i.e., MRZ information in the list used for the derivation of K_{ENC} and K_{MAC}) has been found. If they do not match, the passport stops the BAC process by sending an error message to the attacker, which reiterates the process with the next entry in the list. The success of this attack depends on the length of the list and the time taken per each iteration, as detailed in Section 7.1. Some security countermeasures have been implemented against this type of attack. The French passport, for instance, adds an incremental delay after each failed BAC. This mechanism is tested in Section 7.1.

An additional countermeasure based on time and remanence decay in SRAM (instead of a counter) was proposed in Rahmati et al. [2012]. It basically allows maintenance of a sense of elapsed time without power with the advantage of not requiring any special-purpose hardware, which makes it particularly attractive for resource-constrained environments. In the case of ePassports, it will provide a slightly different way of delaying unauthorized access that is capable of ignoring previous false authentication attempts if the passport has been removed from the reader's range for an appropriate time lapse. This is an interesting alternative that, to the best of our knowledge, has not yet been implemented in any ePassport.

6.1.2. Offline Brute-Force Attack. This attack overcomes the communication rate and processing time issues. The attack is carried out by cracking a pair[10] generated during the session key derivation in a legitimate communication[11], either the one sent by the reader or that sent by the passport. The attack is passive in the sense that the attacker eavesdrops only on the communication. In practice, the command APDU sent by the reader is `00 82 00 00 28 DATA 28` and the APDU answer sent by the passport is `DATA 90 00`, where `DATA` is the 40B pair (32B ciphertext and 8B MAC) to be captured. Once the legitimate pair has been captured, the attacker separates the ciphertext from the MAC. The attacker then performs an exhaustive key search on the MAC generation using every potential key to generate the MAC of the ciphertext until the output matches the MAC from the pair previously captured, which means that the key has been found.

Even though this attack is practical using off-the-shelf eavesdropping devices such as OpenPCD[12] and Proxmark III[13], it requires a specific scenario given that legitimate

[10]A pair is the concatenation of the ciphertext with its MAC.
[11]A legitimate communication is a communication between the passport and a reader that knows the MRZ.
[12]http://www.openpcd.org.
[13]http://www.proxmark.org.

communications usually occur in secure environments such as airports and borders. Also, given that passports are compliant with ISO/IEC 14443, they cannot be eavesdropped at a long distance. Using off-the-shelf reading devices, querying a passport can be done at a distance of a few centimeters (up to roughly 1 meter with an ad hoc device equipped with a large antenna) and the signal can be eavesdropped at a distance of a few meters (depending on whether the forward or backward channel is considered) [Hancke 2011].

Some related analyses have been done on passports from other countries such as The Netherlands (Hoepman et al. [2006] and Pooters [2008]), Germany (Carluccio et al. [2006], Friedrich [2006], and Liu et al. [2007]) and the United States [Juels et al. 2005].

6.2. Response-Time Vulnerability

During the BAC, once the reader sends the first pair, the passport computes the MAC of the ciphertext and compares it to the concatenated MAC. If they do not match, the passport sends an error message to the reader. Otherwise, it decrypts the 3DES ciphertext, extracts the RND'_{ICC} (i.e., from the 8^{th} to the 16^{th} byte) and compares it to RND_{ICC}. If they do not match, the passport sends an error; otherwise, the BAC process continues. Decrypting the 3DES ciphertext and comparing the random numbers takes some milliseconds for the passport. This processing time, which can be easily measured by an attacker, discloses whether the BAC process failed at the MAC comparison stage or at the RND_{ICC} comparison. If it failed at the MAC level, this implies that the reader used the wrong MRZ. If it failed at the RND_{ICC} comparison, this means that the reader generated the MAC with the correct key, but did not use the RND_{ICC} sent previously.

Similar to the offline brute-force attack (see Section 7.1.3), this needs to capture an exchanged pair from a legitimate communication. Once the legitimate pair has been obtained, the attacker can label it in order to remember to whom it belongs. The next time that the attacker sees a passport, the attacker can check if the passport has generated one of the pairs the attacker has captured. Since the processing time is not exactly the same from one passport to another, the attacker first needs to send a pair that the attacker definitely knows to be wrong (e.g., by sending an arbitrary 40B number) and measures the time taken from the beginning of the BAC process until receiving the error message (t_1). Then, the attacker sends the pair captured previously and measures the time again (t_2). By observing the difference in response times, the attacker can easily determine whether the passport under examination is the same one that generated the captured pair.

If $t_1 - t_2 = \delta$ is big enough, this means that the passport took more time to process the second pair, which will likely be due to the deciphering time. Therefore, the MAC verification succeeded, which implies that the passport is the one that generated the second pair. In research undertaken in Chothia and Smirnov [2010], they found that δ should be around $1.7ms$ to get a good trade-off between false positives and false negatives.

This attack requires the same equipment as the offline brute-force attack, thus raises the same issues, that is, it is complicated, risky, and expensive, especially when the final goal is just the identification of the passport, without any possibility of later access to its contents.

Chothia and Smirnov [2010] also figured out that old French passports send a different error message when the BAC failed at the MAC comparison or at the RND_{ICC} comparison. Based on this trivial design mistake, it becomes even easier (and more accurate) to identify the passport.

6.3. Fingerprinting Methods

Fingerprinting methods have been found at different levels of the communication.

6.3.1. Error Message Fingerprinting. The response command defined in ISO/IEC 7816-4 [ISO 2005] contains a 2B long status word that indicates the status of the command sent (e.g., success, wrong length, or file not found). Those have been defined in ISO/IEC 7816, but the ICAO description does not specify the status assigned to each possible error. Therefore, it is up to the passport manufacturer to define what should be answered in case of error. For example, a Belgian passport sends the error message `SW1:0x67 SW2:0x00` (i.e., wrong length L_E) to a *GET CHALLENGE* command with a wrong L_E (`0x09` instead of `0x08`), while a French passport sends the error message `SW1:0x6C SW2:0x08` (i.e., wrong length L_E, where SW2 provides the exact length). Every different response APDU can be used to fingerprint passports [Richter et al. 2008]. This weakness allows an attacker to identify the issuing country among different passports by sending a number of specific command APDUs. It is even possible to identify roughly the date of issue among passports issued from the same country, since the answers may change between versions.

An attacker first needs to create a database with specific APDUs that trigger different responses for different issuing countries and versions. This identification of the issuing country and version might not be possible with a single command APDU. By using a set of carefully chosen APDUs, however, the attacker can narrow down the list of potential countries and versions after receiving answers to each specific command, eventually finding all relevant information.

The main problem with this attack is that, even though the vulnerability does not provide lots of sensitive information (i.e., the issuing country, and maybe the version), it is very easy to exploit and the attacker does not need to know anything about the bearer. A countermeasure would be to standardize the status answers between countries; if done properly, this will offer a long-term solution to this attack.

6.3.2. File Control Information. Upon a valid *SELECT FILE* command, the ICC may reply in some implementations with *success* (`0x90 0x00`) even before BAC (but a read binary would fail), and also may respond with a File Control Information (FCI) [ICAO 2008b]. There are three FCI templates: (i) The File Control Parameters template, (ii) the File Management Data template, and (iii) the File Control Information template, which start with `0x62`, `0x64`, and `0x6F`, respectively. Those templates are BER-TLV[14] data objects. Depending on the issuing country and its version, the passport might send different FCI templates or, more likely, just send an empty response APDU. Usually, whenever passports respond to a *SELECT FILE* with an FCI, they differ depending on the issuer and the version [Richter et al. 2008]. Therefore, it is possible to identify the issuing country (or limit the potential issuers to only a few) among different passports following this method.

6.3.3. Answer to Select and Protocol Info Bytes. As previously mentioned in Section 5.1, passports can be implemented with two different communication signal interfaces, referred to as Type A and Type B. If this single fact allows partitioning of passports into two groups, each of those standards define extra bytes to control the communication parameters. Type A passports send an ATS[15] on a low layer of the communication that may be used for the fingerprinting of a passport [Chothia and Smirnov 2010; Hoepman et al. 2006; Richter et al. 2008]. ICC embedded in passports may slightly differ from

[14]Basic Encoding Rules Type-Length-Value (BER-TLV) is a self-describing and self-delimiting hierarchical encoding format composed of the *tag* (1B), the *byte-length* of the content (1B) and the *content*.
[15]In reply to a *Request for Answer to Select* (RATS).

one version or country to another, as does the ATS. Therefore, it is possible to create passport groups just by examining their ATS. Type B passports send protocol info bytes that can be abused for a similar type of fingerprinting.

6.3.4. Unique Identifier. The UID used for the anticollision mechanism on RFID tags has to be unique in an interrogation zone and readable by the reader without any additional security measures. The UID allows an attacker to identify and track [Hoepman et al. 2006; Pooters 2008; Richter et al. 2008] a passport unless the UID is randomly generated and changes every time it enters an interrogation zone. Even in that case, random UIDs also do reveal information. Indeed, very few RFID tags have random UIDs, thus chances are (at least for now) that, if you encounter one, it will be that of a passport. An additional point is that, according to ISO/IEC 14443 [ISO 2011], a random UID should start with `0x08`, but some passports are not compliant with the standard and use random UIDs without the `0x08` header. This and similar problems have been highlighted in the literature already [Monnerat et al. 2007].

6.3.5. Response Times. In addition to the initial anticollision phase, in which ISO/IEC 14443 [ISO 2011] defines the response times very strictly, responses to an APDU can be sent by a passport at various times below a maximum limit. Response candidates to observe are, for example, responses to RATS, Select MRTD, Select EF.COM, Get Challenge, and so forth. Different ePassport implementations lead to different timing behaviors, with a variety even richer than that observed with error messages. One single ePassport can also show a slight variety of response times due to its countermeasures against side-channel attacks (e.g., by injecting random delays in its process); therefore, it is wise to acquire a few measures on one single passport and compute their average and variance. It is unrealistic to standardize response times because they are competitive arguments for solution vendors, and it would be unacceptable to align everybody on the slowest implementation.

6.3.6. Physical Layer. Avoine and Oechslin [2005] suggested that the physical layer characteristics of an RFID tag leak too much information that can be used for fingerprinting. Later, Danev et al. [2009] practically proved that this approach worked quite well at individually identifying RFID transponders, thus also affecting ePassports. Based on their physical behavior, they achieved an error rate of 2% and 4% only. So far, such identification methods require a physical access to the passport in order to perform multiple measures with sufficiently accurate results; thus, for the moment, they are not one of the major privacy concerns. This could change at any time if some improvement is found that allows these measurements to be performed accurately at a certain distance or with off-the-shelf commercial readers.

6.3.7. Multi-Fingerprinting. As already suggested with the *error message fingerprinting attack*, an attacker can create a database that contains as many fingerprints (e.g., APDU/FCI, ATR, UID, timings) as possible in order to be more accurate, and match them to batches of passports. The fingerprinting methods identify either a family of passports (i.e., a version of a passport from an issuing country) or the passport itself (i.e., individual identification). Thus, each group of passports in the database refers to a COUNTRY + ISSUING DATE for family identification or an identity information for individual identification. Once the attacker has built a database, the attacker can start querying passports for identification by sending the requests needed to get the required fingerprint. If the fingerprint matches an entry in the database, the attacker checks the related data to see the potential families of passports. Note that if some forms of fingerprinting (such as error message fingerprinting) can be deceived by exhaustive standardization and proper implementation, some are impossible to eradicate because they exploit more fundamental implementation choices: type A vs. type B, type A ATS,

type B protocol info, timings depending not only on an ePassport applet but also on the underlying operating system, and so forth. Some error messages are even out of the reins of implementers, as can be seen in the case described in R12-TR_SAC_0005 in ICAO [2013b], in which ICAO was forced to tolerate that a Select before BAC can return *success* (0x90 0x00) instead of *security status not satisfied* (0x69 0x82) due to some ICC operating system constraints. Fingerprinting methods are also very useful to reduce the key space to be explored in the online brute-force attack described in Section 6.1.

6.4. MRZ Lookup Table

A vulnerability presented in Sportiello [2012] targets old generations of Italian passports. By manipulating the *GET CHALLENGE* command, it is possible to force the passport to send a challenge with the seven less significant bytes set to 0x00. Sportiello figured out that sending a *GET CHALLENGE* command with the L_E set to 0x01 (L_E is the expected response length and should consequently be equal to 0x08) forces the passport to send only the first byte of the challenge RND_{ICC} it generated, while internally considering that the other seven less significant bytes are left initialized to 0x00. Sportiello discovered that some Italian passports actually generate a challenge between 1B and 8B long, depending on the value of L_E, and padded it with 0x00 to reach 8B, the valid challenge length for the BAC process.

The goal of the attack (Figure 9) is to find the MRZ, based on a ciphertext generated during the BAC and a lookup table. The attack requires eavesdropping on a legitimate BAC (see Section 3.1), in which the passport generates a challenge with 8B set to 0x00. Whenever the passport sends the K_{ICC} to the reader, the first 8B of the ciphertext[16], are used for the lookup of the MRZ.

The first step is to establish a Man in the Middle (MITM) attack[17] between the reader and the passport. When the reader wants to access the content of the passport, it starts with the BAC: The reader sends a *GET CHALLENGE*, and the MITM attacker intercepts the command and sets L_E to 0x01. The modified command is sent to the passport until the latter answers a 1B challenge set to 0x00. In such a case, the passport internally considers that the challenge RND_{ICC} is an 8B challenge set to 0x00^8.

The null 8B challenge is then forged and sent back to the reader. As usual, the reader responds with a ciphertext (E_{IFD}) and a MAC (M_{IFD}) calculated from RND_{IFD}, RND_{ICC}, and K_{IFD} (see Section 3.1 for more details about the BAC). The command is forwarded to the passport, which performs the usual verification. Finally, the passport sends a ciphertext (E_{ICC}) and a MAC (M_{ICC}) calculated from $\text{RND}_{\text{ICC}} = \{\text{0x00}\}^8$, RND_{IFD}, and K_{ICC}. The ciphertext is intercepted by the MITM and the first 8B are used in the lookup table to find the matching MRZ.

Such lookup is possible because the ciphertext is a known plaintext encrypted with 3DES in CBC mode with an $IV = 0$ and a key K_{ENC} derived from the MRZ. Thus, the lookup table is built by encrypting $\{\text{0x00}\}^8$ with the entire MRZ key space.

6.5. Early Active Authentication

For AA (see Section 3.3), the reader verifies that the passport is not a counterfeited one by sending a challenge that the passport signs with its private key located in a secured memory. The reader verifies the signature thanks to the public key located in DG15. This procedure is supposedly performed after the BAC and the PA, in order to ensure the validity of the public key. However, there exist passports that accept to execute the

[16] $\text{ENC}_{K_{\text{ENC}}}(\text{RND}_{\text{ICC}} \| \text{RND}_{\text{IFD}} \| K_{\text{ICC}})$.

[17] An MITM attack requires the attacker to be able to block, tamper with, and generate messages exchanged between the reader and the passport.

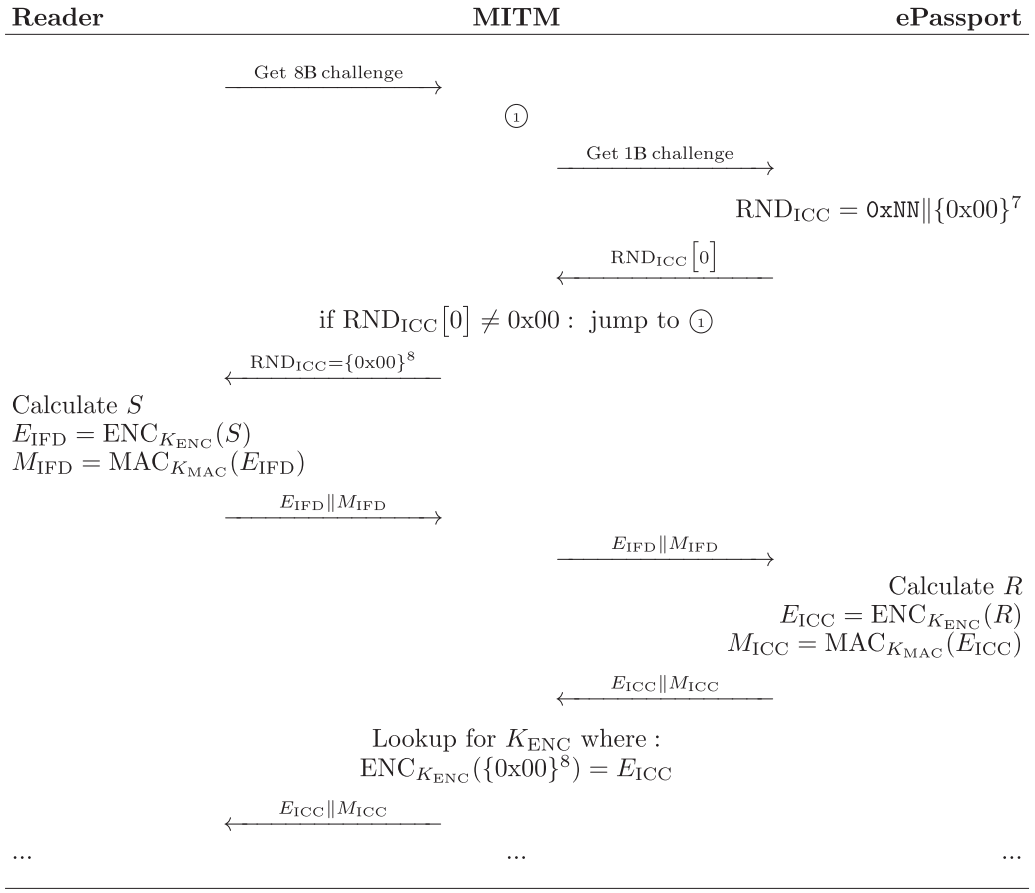

Fig. 9. MRZ lookup table attack process.

AA before the BAC and PA. This allows attackers to easily get a signature from the passport calculated on 64b of data chosen by the attacker and random data generated by the passport.

6.5.1. Signature Evidence. Hoepman et al. [2006] describe a way to exploit the security weakness described earlier: instead of sending a random number, the attacker sends 64b of meaningful data. For instance, an attacker is able to build evidence of meeting a given person (without that person's awareness) by sending to the person's passport an 8B challenge that contains information from the last newspaper. The signature received from the passport is proof that the attacker met the victim after the publication of the newspaper.

6.5.2. Traceability Attack. A vulnerability has been briefly described in Juels et al. [2005], who point out that a traceability attack can be done by analyzing the signature generated by the passport during the AA. When RSA is used, the signature s is actually generated by signing the message m (see Section 3.3) as follows: $s = m^d (\mathrm{mod}\ n)$. The signature is then verified by calculating $m = s^e (\mathrm{mod}\ n)$. Each passport has a unique private exponent and modulus. Therefore, the modulus may be used as an identifier for traceability purposes, as mentioned in Bellare et al. [2001] and Desmedt [1995]. Due to the modulo operation, a signature will never be larger than $n - 1$. Running the

Table IV. Average Residual Delta Between the Modulus n and the Highest Observed Signature

Number of executions of AA	10	50	100	500
Residual delta	9.09%	1.96%	0.99%	0.20%

Fig. 10. ePassport viewer main interface.

AA with a random $\mathrm{RND}_{\mathrm{IFD}}$ for a while, and storing the highest signature value seen (s_{MAX}) should end up with an s_{MAX} value very close to the modulus n. Assuming that the RSA algorithm provides a uniformly distributed signature, the residual delta $\frac{n-s_{\mathrm{MAX}}}{n}$ is, on average, equal to $\frac{1}{N+1}$ after N executions of AA. Table IV provides the theoretical residual delta between the targeted modulus n and the highest observed signature.

The goal of the attacker is to run as many AAs as needed to reach the expected accuracy. Once s_{MAX} is computed, the attacker keeps it as an identifier of the passport. Whenever the attacker wants to identify a passport, he proceeds in the same way to get the highest signature. Then, the attacker computes the difference between the reference, that is, the identifier, and the new high signature (see Table IV). The attacker checks if the two signatures are close enough to identify the passport as the same. The more AAs run, the less likely it is that this attack will trigger false positives and false negatives.

Even though this scheme requires the attacker to be close to the passport to capture enough signatures, the fact that the attacker does not need any legitimate communications, and only a passport that allows the AA to take place before the BAC, makes it very practical.

7. EXPERIMENTAL RESULTS

This section illustrates the vulnerabilities introduced in Section 6, applied on real passports. The experiments are performed using an extended version of the software ePassport Viewer[18], illustrated in Figure 10. In spite of its name, ePassport Viewer is a powerful software that offers the possibility to launch various attacks on ePassports. The software is used with an SCM SCL3711 RFID reader[19].

7.1. MRZ Low Entropy

We have performed an MRZ brute-force attack on a limited subset of possible MRZs. This attack aims to evaluate the time needed to crack an MRZ. In the current (highly optimistic) scenario, we assume that the attacker knows the date of birth of the holder of the passport under attack, the date of expiry and, thanks to the latter information,

[18]http://code.google.com/p/epassportviewer/.
[19]http://www.identive-group.com.

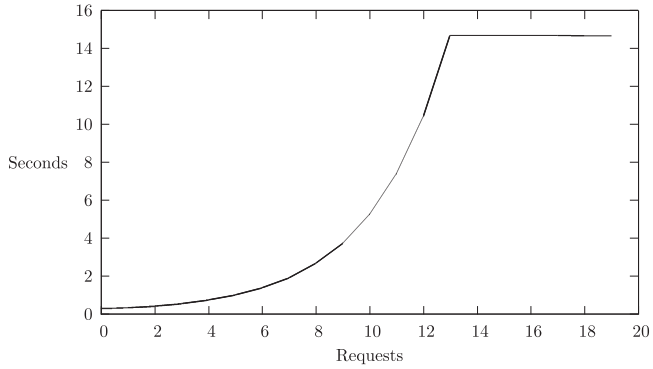

Fig. 11. Response time of a 2011 French passport after N failed BACs.

the attacker knows that the passport number belongs to a range of 600 possible values. The measured times allow for extrapolating the expected time needed to perform the attack on the full set of MRZs.

7.1.1. Online Brute-Force Attack. Testing one MRZ takes approximately 23ms on a Belgian passport, which means that an attacker is able to test about 43.5MRZs per second. With an entropy of 2.28×10^{10}, that is, 34.41, (see Table III), the attack would still take about 16.5 years to scan all possible MRZs. This highlights the nonpractical aspect of this attack if the attacker has no information about the passport or its holder. Under more favorable circumstances, however, this delay can be drastically reduced, as shown in the attack performed by Adam Laurie [Kirk 2007]: UK passports can be delivered by regular post in a sealed envelope so that the postmark gives a pretty good idea of the expiration date (and of the document number) and the name on the envelope easily leads to a date of birth after an Internet search. The key was cracked after about 40000 attempts, which can be converted into approximately 15.28b of entropy in a matter of minutes.

7.1.2. Online Brute-Force Attack with Countermeasures. The latest Belgian passport implements a security mechanism that, once an external authentication fails, returns the error `SW1: 0x69` and `SW2: 0x82` (i.e., security status not satisfied) to each next external authentication until the connection is reset. The attacker must consequently reset the connection after each unsuccessful attempt in order to pursue the brute-force attack. This slows down the attack to an average rate of 11.5 MRZ/s with our experimental setting.

French passports have another protection against brute-force attacks, based on the use of a counterlike mechanism, as suggested in Avoine et al. [2008]: every time a BAC fails, the time needed to make another BAC is increased up to a certain limit. We performed several experiments on the Basic Access Control of a French passport to test this. We noticed that, once a BAC execution fails (using a wrong MRZ), the time taken by the passport to answer to the next Mutual Authenticate command (i.e., the command used in BAC) increases. The delay actually increases up to approximately 14s after 14 unsuccessful executions. At this point, the response time remains 14s as long as BAC executions fail. Figure 11 represents the response time of a passport[20] after multiple

[20]The experiment has been performed on a French passport issued in 2011 with an Omnikey 5321 Reader. For each value x, the experiments have been performed 9 times. A similar experiment on a French passport issued in 2010 provided results consistent with these. We observed that French passports issued before 2008 do not benefit from this mechanism.

Table V. Response Time of a 2011 French Passport After Failed BACs

N^{th} BAC attempt	Average time in seconds	Standard deviation
1	0.07233	0.00760
2	0.13617	0.02658
3	0.23085	0.05174
4	0.36271	0.08855
5	0.54720	0.13916
6	0.81165	0.21294
7	1.18086	0.31569
8	1.70573	0.46132
9	2.44755	0.66820
10	3.49909	0.95847
11	4.98393	1.36918
12	7.08061	1.94963
13	10.05095	1.53667
14	14.24826	0.94271
15	14.25046	0.94138
16	14.25043	0.94218

consecutive BAC executions with a wrong MRZ. Table V provides the average timing values observed and the standard deviation. Note that, when the passport enters into this kind of "protecting mode," it stays in it until a correct MRZ is provided. Removing the passport from the reader's field, even for several days, does not change this behavior.

7.1.3. Offline Brute-Force Attack. The offline brute force attack requires the attacker to capture a ciphertext during a successful BAC execution. The passport indeed accepts pursuing the BAC execution if and only if it has been able to successfully authenticate the reader. As a consequence, the offline brute-force attack cannot be executed against a passport without a legitimate reader. Once the attacker eavesdrops on a valid ciphertext, the attacker tests every MRZ until finding a match between the eavesdropped ciphertext and the one calculated from the tested MRZ. Given that the brute-force is executed offline, it is much faster than the online attack previously described. In the setting described earlier (i.e., considering a set of 600 different MRZs), the attack takes around 0.18s, on average, using ePassport Viewer (Figure 12) to find the correct MRZ value.

7.2. Response-Time Vulnerability

To test whether a passport is vulnerable to timing measurements, the attacker first eavesdrops on a legitimate BAC execution and records the valid pair ciphertext/MAC (Step 1). Afterwards, the attacker performs a BAC with an arbitrary MAC value and measures the timing (Step 2). Finally (Step 3), the attacker performs another BAC using the valid pair ciphertext/MAC obtained in Step 1: if the response time is significantly different from the one measured in Step 2, then the passport is vulnerable. Otherwise, the attacker cannot access the passport.

We implemented this attack in ePassport Viewer and applied it to a French passport issued in 2011. Sending a message with a wrong MAC produced a "No information given" message (`0x90`, `0x00`) and a response time equal to 42ms. The second BAC (using an arbitrary MAC value) produced the same error message, with a response time equal to 56ms. Based on the difference between the two response times, we can conclude that this passport is highly likely to be vulnerable to this attack.

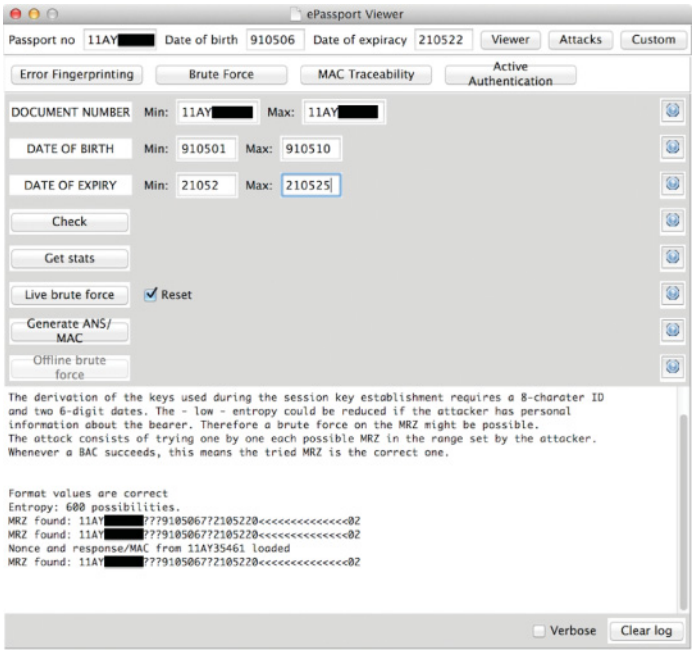

Fig. 12. ePassport Viewer brute force attack.

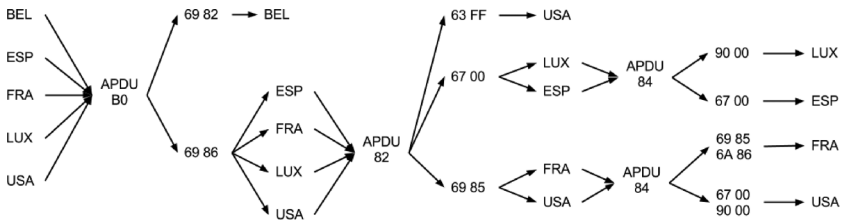

Fig. 13. Passport fingerprinting strategy. Passports are tested from left to right.

7.3. Fingerprinting Methods

We apply in this section several fingerprinting techniques against ePassports. Suggestions to stop or mitigate the impact of these fingerprinting techniques are offered along the section and in Section 8.

7.3.1. Error Message Fingerprinting. The aim of the error message fingerprinting attack is to identify which country issued the targeted passport, and possibly which generation the passport belongs to, by exploiting error messages returned by the passport upon reception of *malformed* commands.

For example, we applied the attack on French passports: modifying the "LE" field of the GET_CHALLENGE APDU from `0x08` to `0x09`, the answer received from a French ePassport issued in 2008 was `0x67 0x08` ("Wrong length") while the answer received from a French ePassport issued in 2011 was `0x6C 0x08` ("Wrong Le field: `0x08` is the exact length").

Applying the error message fingerprinting attack with several different commands on a large batch of passports allows the attacker to discover the countries that issued the passports. Figure 13 illustrates how six passports from different countries with

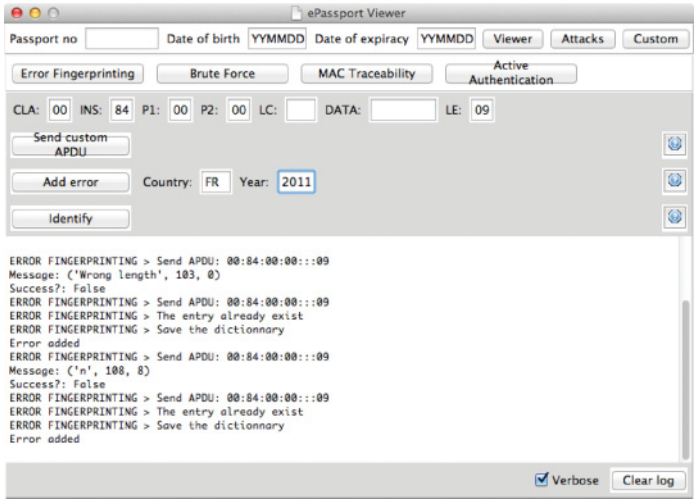

Fig. 14. ePassport viewer error fingerprint attack.

different years of issue can be discriminated. An APDU `0xB0` (Read Binary) allows an attacker to immediately identify a Belgian passport. An APDU `0x82` (External Authentication) allows the attacker to identify one of the two American passports, and split the four passports left into two clusters. Finally, an APDU `0x84` (Get Challenge) discriminates all the remaining passports.

The ePassport Viewer software maintains a collaborative database of observed error messages, which allows attackers to easily identify nationalities of targeted passports, as shown in Figure 14.

7.3.2. Answer to Reset. As explained in Section 5.2.1, PCSC-compliant readers forge an ATR encapsulating passport Type A ATS or Type B *Application Data* and *Protocol Information* in its *historical bytes*. Therefore, those ATRs always have the following structure: $3B \| 8n \| 80 \| 01 \| historical\_bytes \| TCK$, where n is the number of historical bytes and TCK is a checksum byte[21].

The ATR is of significant value, because it allows the identification of a passport family and potentially to focus on the number of security vulnerabilities and exploits associated with it. Table VI summarizes ATRs for a few countries and years of issue whenever they are known. Some of the ATRs were found in smartcard_list.txt[22], an impressive list of ATRs maintained for years by Rousseau, some of which were collected by the authors (and contributed back to that list). It is interesting to see that if ATRs allow isolation of batches of passports from same date and same country, occasional collisions may occur across several countries, as highlighted in the table. Note also that recent Spanish passports use the smallest possible ATR because their ATS does not contain any *historical byte* and therefore does not leak any information.

7.3.3. Unique Identifier. UID is an element specific to ISO/IEC 14443 Part 3 and absent from ISO/IEC 7816; therefore, to get that information through a PC/SC compliant reader, one has to send a pseudo-APDU defined in PC/SC Part 3, naturally called "GetUID." The difference with regular APDUs is that this pseudo-APDU is intercepted

[21]XOR of all the bytes from second byte $T0 = 8n$ to end of historical bytes.
[22]http://ludovic.rousseau.free.fr/softwares/pcsc-tools/smartcard_list.txt.

Table VI. Country and Date of Issue According to ATR Value

ATR	Country	Date of Issue
3B 80 80 01 01	ESP	2012
3B 84 80 01 00 00 90 00 95	LUX	2012
3B 84 80 01 04 38 33 B1 BB	NLD	unknown
3B 85 80 01 80 73 84 21 40 12	NLD	unknown
3B 88 80 01 00 00 01 07 01 72 90 00 EC	BEL	2009
3B 88 80 01 E1 F3 5E 11 73 81 A5 00 03	USA	2007
3B 88 80 01 E1 F3 5E 11 77 81 A5 00 07	USA	2012
3B 88 80 01 E1 F3 5E 11 77 81 C7 20 45	FRA	2007-2008
3B 88 80 01 E1 F3 5E 11 77 83 95 00 35	FRA	2012
3B 88 80 01 E1 F3 5E 11 77 83 D5 00 75	DEU	2009
3B 89 80 01 00 64 04 15 01 02 00 90 00 EE	DEU	2007
3B 89 80 01 4A 43 4F 50 34 31 56 32 32 4D	NZL	unknown
3B 89 80 01 4D 54 43 4F 53 73 02 01 04 3A	CZE	unknown
3B 89 80 01 4D 54 43 4F 53 73 02 01 04 3A	LBY	unknown
3B 89 80 01 80 67 04 12 B0 03 05 01 02 4C	AUT	unknown
3B 8B 80 01 00 31 C0 64 B0 FC 10 00 00 90 00 53	THA	2010-2011
3B 8B 80 01 00 64 04 11 01 01 31 80 00 90 00 5A	LUX	2007
3B 8B 80 01 00 64 04 11 01 01 31 80 00 90 00 5A	DEU	2006
3B 8B 80 01 00 64 04 11 01 01 31 80 00 90 00 5A	GBR	unknown
3B 8C 80 01 4F 54 49 44 28 94 B3 C0 01 00 90 00 45	BEL	2009-2013
3B 8C 80 01 4F 54 49 44 28 94 F7 C0 00 00 90 00 00	FRA	2010-2011, 2013
3B 8C 80 01 50 18 61 88 CE E1 F3 5E 11 77 81 C7 0E	FRA	2007, 2008
3B 8C 80 01 80 91 E1 65 D0 00 43 00 00 82 90 00 19	FRA	2008
3B 8C 80 01 80 91 E1 65 D0 00 46 00 00 82 90 00 1C	USA	2009
3B 8D 80 01 80 91 E1 65 D0 00 5B 01 03 73 D4 41 40 B6	FRA	2010-2011
3B 8E 80 01 0E 78 33 C4 02 00 64 04 15 01 02 00 90 FF 95	ESP	unknown
3B 8E 80 01 10 38 77 A7 80 91 E1 65 D0 00 42 00 00 82 72	CZE	2009
3B 8E 80 01 80 91 91 31 C0 64 77 E3 03 00 83 82 90 00 1C	BEL	2005
3B 8E 80 01 80 91 91 31 C0 64 77 E3 03 00 83 82 90 00 1C	THA	2005
3B 8E 80 01 80 91 E1 31 C0 64 77 E3 03 00 83 82 90 00 6C	BEL	2006-2007

by the reader or its driver who know already the UID and reply immediately without additional communication with the passport. The UID identifies the type of passport to a certain extent. The vast majority of ePassport UIDs are random and start with `0x08`, meaning that, according to ISO/IEC 14443, the UID is randomly generated. There are, however, a few exceptions; for example, earlier French and Belgian passports used a static UID and Australian ones had UIDs made of four random bytes, including the first one, which is not `0x08`, contrary to the standard recommendations. The mere fact that most ePassports use a random UID allows distinguishing them from other RFID devices because random UIDs are rarely used elsewhere.

Table VII summarizes the results of our experiments performed on a large batch of passports. The type of UID (random/static) is provided according to the country and date of issue.

7.3.4. Response Times. We did some experiments with response time–based fingerprinting, which can be presented with the help of star plots. Passport observations have been split into a *slow* group (Figure 15) and a *fast* group (Figure 16) for better readability. Each star plot consists of a sequence of four spokes representing response times to *RATS*, *Select MRTD*, *Select EF.COM* and *Get Challenge* commands. The data

Table VII. UID Type According to Country and Date of Issue

Country	UID type	Date of issue
BEL	static	2006
BEL	random	2007-2012
ESP	random	2012
FRA	static	2007-2008
FRA	random	2009-2013
LUX	random	2007-2012
USA	random	2007-2012

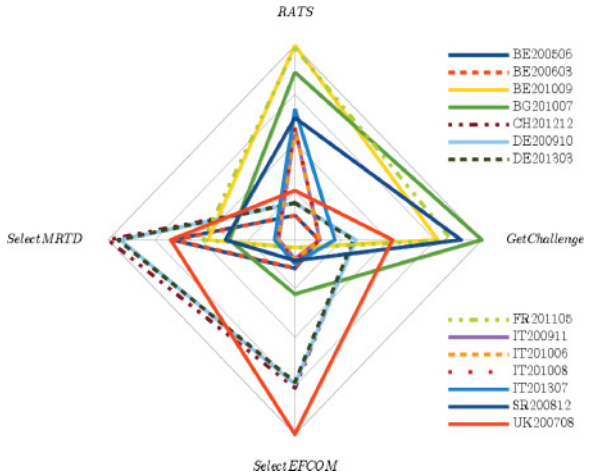

Fig. 15. Normalized star plot of response times to *RATS*, *Select MRTD*, *Select EF.COM* and *Get Challenge* commands, group of *Slow* passports.

length of a spoke is proportional to the magnitude of the variable for the data point relative to the maximum magnitude of the variable across all data points. Those maximum magnitudes are the slowest responses observed and summarized in Tables VIII and IX. A line is drawn connecting the data values for each spoke such that each observed passport is represented by a polygon. Star plots help detect clusters of passports or unexpected observations [NIST/SEMATECH 2013].

Measures of passports based on the same implementation perfectly match, such as IT200911 ≅ IT201006 ≅ IT201008, DE200910 ≅ DE201303, BE200506 ≅ BE200603, and BE200812 ≅ BE200901, but also, more surprisingly, BE201305 ≅ FR201303 or the Swiss and German passports, which seem to share a very similar, if not identical, implementation. The FR201307 curve has some characteristics deserving more explanation: it is of type B, and therefore does not handle *RATS*, thus the horizontal line; it is pretty fast, but from time to time any of the responses is delayed by a constant amount; therefore, the second triangle FR201307* showing the slower answers. One reason for such behavior could be an asynchronous process such as an entropy collector being called from time to time when the entropy pool gets empty, but this is pure speculation.

Note that brute-force countermeasures implemented, for example, in the tested French and Dutch passports, were not activated because we stopped at the *Get Challenge* command before attempting to perform a real authentication.

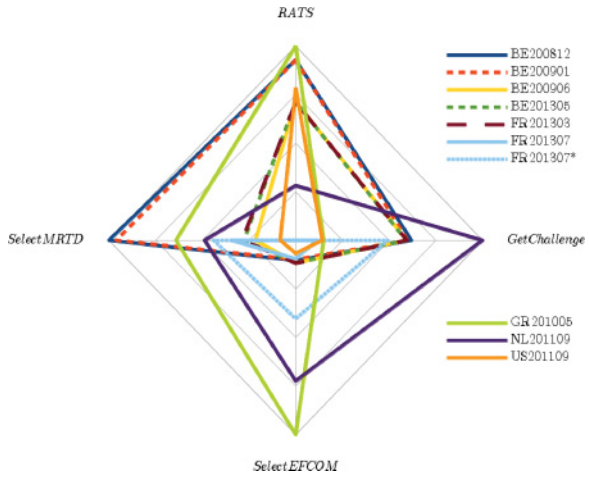

Fig. 16. Normalized star plot of response times to *RATS*, *Select MRTD*, *Select EF.COM* and *Get Challenge* commands, group of *Fast* passports.

Table VIII. Slowest Response Times, Group of *Slow* Passports

Command	Response time	Passport
RATS	588μs	BE201009
Select MRTD	229.65ms	CH201012
Select EF.COM	51.93ms	UK200708
Get Challenge	24.42ms	BG201007

Table IX. Slowest Response Times, Group of *Fast* Passports

Command	Response time	Passport
RATS	262μs	GR201006
Select MRTD	43.97ms	BE200812
Select EF.COM	13.44ms	GR201006
Get Challenge	11.00ms	NL201109

Those measures were not computed with ePassport Viewer, but by a small program based on `libnfc`-timed commands, to achieve greater accuracy without any influence from PC/SC or USB stacks.

7.3.5. Physical Layer. This attack is quite hard to implement because some very specific and precise (and costly) hardware is needed. The goal of this attack is to identify the passport at the physical layer by listening to the exchanges between a reader and a passport. Danev et al. [2009] concluded that passports can be uniquely identified due to multiple variations at different stages of the manufacturing process.

7.3.6. Summary of Fingerprinting Methods. With all these fingerprinting techniques available, it is possible to define a method built upon the various results obtained. With just a few command APDUs, it is possible to accurately identify the country and the generation of a large number of passports. By aggregating the other fingerprinting options available (ATR, UID, and so forth), the identification of a passport is even faster and more precise.

These fingerprinting attacks and their undesirable consequences on privacy can be mitigated, if not avoided. First, a metal cover that physically stops any passport reading if not open—as already used in the United States passport—effectively creates

a Faraday cage. A second approach consists of a better, more uniform, and strict implementation of the standards so that all compliant passports must provide the same outputs to the same inputs. The latter approach presents some inherent limitations that make the former particularly fit for purpose.

7.4. Passive Authentication Weaknesses

PA is actually nothing more than a digital signature and is consequently a secure concept, assuming that it uses secure algorithms, random challenges and keys, and so forth. The major weakness of digital signatures in large-scale applications is the difficulty of verifying the authenticity and integrity of the root certificates, which are disseminated. If authenticity and integrity of the root certificates cannot be verified, anyone who uses digital signatures lives in a state of sin. This problem was illustrated with ePassports when the famous hacker group THC demonstrated that a reader located in Schiphol Airport in Amsterdam accepted as valid a passport containing a picture of Elvis Presley. Although the reader was installed in the airport for demonstration purposes only, a major question was raised by the media, that is, whether an official reader would have detected the fake passport.

This security problem due to the poor diffusion of the CSCA certificates justified the need to set up an ICAO PKD [Central Intelligence Agency 2011]. However, in 2013, the ICAO PKD counted only 37 participants among more than 100 countries issuing ePassports [ICAO 2013a]. Roman Vanek mitigates the impact saying that "74% of the ePassports issued so far have been issued by a PKD participant." This nevertheless means that 26% of issued passports are probably not checked when they cross borders and, even worse, that more than 150 countries all over the globe are not PKD participants (a problem especially acute in South America, South Africa, and the Middle East). In light of this, in theory, there could be around 500 million passports that are never fully security checked during border controls. This observation is probably overstated because countries can have bilateral agreements, and being a PKD participant is not a requirement to get access to the PKD: nonparticipants can indeed have access to the PKD, but through a web interface only.

An analysis of the data available from ICAO provides an accurate picture of the current situation in terms of CSCA certificates, DS certificates, and revoked DS certificates.

7.4.1. CSCA Certificates. Several issues threaten the use of the CSCA certificates. A major issue is the fact that countries sometimes change the issuer name of their certificates, for example, "OU=MOFA" in Korean passports became "OU=MOFAT" later on. It is disconcerting as well to see "OU=Singapore Trial Passport" in a certificate issued in July 2006.

Another observation can be made regarding the validity period of the private key associated to a certificate. Indeed, the validity period of a CSCA certificate should be the validity period of the associated CSCA private key extended by the validity period of a DS certificate (which, in turn, should be the validity period of the associated DS private key extended by the validity period of a passport). Typically, it should be 3 years (validity of a CSCA private key) + 4 months (validity of a DS private key) + 5 or 10 years (validity of a passport). Many countries do not take this calculation into account at all, and assign the same *not-after* date to both CSCA certificate and CSCA private key. Other countries simply discard the validity period of the private key because this information is not mandatory in the certificates. Other countries have surprisingly long validity periods, for example, 15 years in certain Austrian certificates.

Table X illustrates the distribution of key types and sizes of CSCA certificates and Link Certificates available in the ICAO PKD[23]. There are 20 certificates using ECDSA

[23]Master Lists v. 0015.

Table X. Signature Schemes Used by CSCA Certificates

Signature	Number of certificates
RSA-2048	4
RSA-3072	30
RSA-4096	90
ECDSA-256	14
ECDSA-384	6
Total	144

Table XI. RSA Exponents in CSCA Certificates

RSA Exponent	Number of certificates
3	9
38129	2
43459	1
65537	112
Total	124

Table XII. Signature Schemes Used by DS Certificates

Signature	Number of certificates
RSA-1024	8
RSA-2048	3120
RSA-3072	2
ECDSA-224	31
ECDSA-256	40
Total	3201

and 124 certificates using RSA. Countries using ECDSA whose CSCA certificate belongs to ICAO PKD are: Cyprus, Germany, Lithuania, Latvia, Russian Federation, Switzerland, and United Arab Emirates.

Table XI shows the RSA exponents used in the CSCA certificates and Link Certificates available in the ICAO PKD[24].

7.4.2. DS Certificates. Countries use different practices to generate DS certificates. For example, most countries renew their DS certificates every 4 months, which follows the ICAO recommendations. The vast majority of countries generate only one DS certificate at a time. Others generate more than one DS certificate at a time; for example, UK generates 58 certificates every 4 months, which implies that they have more than 1750 certificates in the ICAO PKD, of which 724 are revoked. As previously done with CSCD certificates, Table XII and Table XIII detail which signature schemes are used, and which exponents are used with RSA. Note that DS certificates for the *United Nations* use RSA-1024 only. All other entities use at least 2048b for their DS private keys.

7.4.3. Certificate Revocation Lists. The analysis of the certification Revocation Lists (CRLs) is also interesting. Table XIV provides the number of revoked keys per country. One may observe that countries are very reluctant to disclose the reasons for the revocation.

[24]Master Lists v. 0015.

Table XIII. RSA Exponents in DS Certificates

RSA Exponent	Number of certificates
3	191
44591	1
65427	1
65537	2937
Total	3130

Table XIV. Revoked DS Certificates

Country	Nb Revoked Certificates	No Reason	Key Compromise	Superseded	Cessation	Unspecified
New Zealand	2	2				
Malaysia	3		3			
AU	4	1		2	1	
USA	3	1	2			
UK	724					724
Canada	15	7			8	
UNO	3	3				
Singapore	6	3			3	
Total	760	17	5	2	12	724

7.4.4. Defect Lists. According to BSI [2010], a "defect is defined as a production error affecting a large number of documents." For example, a defect can be a wrong calculation of a hash value during the personalization process of a passport. Defect lists are signed by the CS certificate that produced the defective documents. Defect lists not only identify the defective passport, but they also allow inspection systems to know how to deal with such passports. For example, a list can provide a replacement certificate when the original DS certificate cannot be decoded properly [BSI 2010, 2013]. As raised in Frontex [2011], if a specific defect implies that the inspection systems should ignore the outcome of a given check for a set of passports, an attacker might exploit this weakness to make passport counterfeiting easier.

7.5. Summary

We have presented in this work the most detailed theoretical and practical discussion of the many security issues regarding ePassports published to date. This includes all previously known weaknesses and some new ones. We have analyzed in depth their real impact on the overall security of the ePassport scheme, and offered multiple suggestions on how to address these problems. With this work, we hope to contribute to safer future standards and implementations.

8. RECOMMENDATIONS

Based on the findings presented in this work, a list of recommendations is provided here. We believe that these recommendations, if adopted and enforced by the ICAO, will significantly increase ePassport security worldwide. This list of concrete recommendations needs to be considered in addition to the more general one offered in the

pioneer work on ePassport security by Meingast et al. [2007], and to the brief set of recommendations offered in Kosta et al. [2007].

(1) *Increase MRZ entropy:* Efforts should be made in at least two directions. First, to limit the heavy formatting that has been observed in some state's implementations of the passport number, which significantly reduces its associated entropy. Second, to actively put in place printing and distribution practices to decorrelate these MRZ values, and avoid cases such as the well documented high correlation between Belgian passport numbers and issue/expiration dates. It is important to note that this will still remain an important issue after the introduction of the SAC, particularly against online brute-force attacks, as SAC will always accept the MRZ as key. Thus, to avoid, for example, the attack of Adam Laurie, your only defense is to significantly increase the MRZ entropy and the delay between attempts (see Recommendation 6).
(2) Implement RF-blocking materials, such as the metallic mesh in the United States ePassports. This is both a very simple and relatively inexpensive security measure that works effectively and stops most unauthorized reading attempts.
(3) The use of SHA-1 in the key derivation process should probably be substituted by the use of bcrypt [Provos and Mazières 1999], PBKDF2 [Kaliski 2000], or other similarly slow key derivation functions specially conceived against brute-force attacks.
(4) In light of the recent Edward Snowden revelations about the United States National Security Agency (NSA) cryptological capabilities, all public key algorithms should significantly increase their key length. In particular, RSA-1024 and ECDSA using curves with nonverifiable random parameters should be abandoned.
(5) Enforce the adoption of incremental delays after failed attempts, as implemented in the French passport. This can also be applied to SAC and any other authentication method put in place.
(6) Minimize response-time vulnerabilities by standardizing random delays in ePassport responses.
(7) Reduce ePassport fingerprinting vulnerabilities by removing entirely *historical bytes*, following the Spanish example.
(8) Reduce ePassport fingerprinting vulnerabilities by adopting standard error and status answers.
(9) Stop the possibility of ePassports performing the AA before the BAC and the SAC. Using AA should actually be avoided (which is the case in several countries, e.g., Germany) given the attacks mentioned in this article. AA should be replaced by a certificate-based authentication mechanism.
(10) Improve the diffusion of the CSCA certificates by making participation in the ICAO PKD mandatory for countries issuing ePassports.
(11) Introduce uniform validity periods for CSCA certificates, set these to relatively short periods (avoiding cases similar to the 15 years of Austria), and enforce its usage.
(12) Regarding the management of Certificate Revocation Lists, more transparency and open disclosure of both issues and good practices would be highly beneficial.
(13) Use only cancellable or revocable biometric data in ePassports to minimize the impact of a security compromise. Do not use raw biometric data, which is inherently nonrevocable. This recommendation is not related to the ePassport application only, but rather applies to any security mechanism that is based on biometric data.

We want to point the reader to an additional interesting proposal in Pasupathinathan et al. [2008], who suggest a novel online authentication mechanism that, to some extent, offers an alternative way of addressing some of the weaknesses previously described.

9. CONCLUSIONS

Although this article provides a long and quite detailed analysis of the multiple security shortcomings in ePassport protocols, considering them globally insecure is an oversimplification. The problem presented to ICAO is complex, with different countries following slightly dissimilar policies at various speeds, using separate manufacturers, even when they fully comply with the standards. It should be noted that, to this extent, ICAO has performed an excellent job in devising secure solutions and updating them as a response to new attacks and vulnerabilities, soon after they were discovered. As is very frequently the case, most of the security weaknesses are not based on the standard itself but on the sometimes inconsistent national implementations of it. ICAO indeed introduced multiple security features but made only a few mandatory to ease the wider acceptance of the standard. This laxity resulted in countries implementing bad security systems in their passports for convenience reasons, misjudgments, or both.

This article shows a general view of the past and current challenges related to ePassports, and aims to inform newer versions of the standard to reduce to a minimum the chances for vulnerable standard implementations, thus helping in a more robust solution for the future. Secure ePassports would provide us with many benefits, in addition to safely crossing borders, and may serve in the near future as a better basis for safer and more trustworthy electronic commerce.

APPENDIX 1

This appendix contains acronyms about passports and cryptography used along the article.

AA	Active Authentication	IV	Initialization Vector
AID	Application IDentifier	LDS	Logical Data Structure
APDU	Application Protocol Data Unit	MAC	Message Authentication Code
ATR	Answer to Reset	MITM	Man in the Middle
ATS	Answer to Select	MRTD	Machine Readable Travel Document
BAC	Basic Access Control	MRZ	Machine Readable Zone
BER	Basic Encoding Rules	OCR	Optical Character Recognition
CA	Chip Authentication	PACE	Password Authenticated Connection Establishment
CAN	Card Access Number		
CLA	CLAss	PA	Passive Authentication
CRL	Certification Revocation List	PC/SC	Personal Computer / Smart Card
CSCA	Country Signing Certificate Authority	PGP	Pretty Good Privacy
CVCA	Country Verification Certificate Authority	PIX	Proprietary application Identifier eXtension
DES	Data Encryption Standard	PKCS	Public-Key Cryptography Standards
DG	Data Group	PKD	Public Key Directory
DSA	Digital Signature Algorithm	PSS	Probabilistic Signature Scheme
DS	Document Signer	RATS	Request for Answer To Select
DV	Document Verifier	RFID	Radio Frequency IDentification
EAC	Extended Access Control	RID	Registered Application provider Identifier
ECDSA	Elliptic Curve Digital Signature Algorithm	RND	RaNDom
EF	Elementary File	RSA	Rivest – Shamir – Adleman
FCI	File Control Information	RSASSA	RSA Signature Scheme with Appendix
FID	File IDentifier	SAC	Supplemental Access Control
FIPS	Federal Information Processing Standards	SHA	Secure Hash Algorithm
ICAO	International Civil Aviation Organization	SOD	Document Security Object
ICC	Integrated Circuit Card	SSC	Send Sequence Counter
IFD	InterFace Device	TA	Terminal Authentication
INS	INStruction	TLV	Type-Length-Value
IS	Inspection System	UID	Unique IDentifier

ACKNOWLEDGMENTS

The authors are grateful to Thomas Sloan and David Barnes for carefully reading previous versions of this work and providing us with useful and interesting insights that have greatly improved it. The authors would also like to kindly thank the anonymous reviewers who provided highly valuable comments.

REFERENCES

American Bible Society. 1999. *Holy Bible: King James Version*. Chapter Nehemiah 2:9.

Gildas Avoine, Kassem Kalach, and Jean-Jacques Quisquater. 2008. ePassport: Securing international contacts with contactless chips. In *Financial Cryptography and Data Security – FC'08 (Lecture Notes in Computer Science)*, Gene Tsudik (Ed.). Vol. 5143. IFCA, Springer, Berlin, 141–155.

Gildas Avoine and Philippe Oechslin. 2005. RFID traceability: A multilayer problem. In *Financial Cryptography – FC'05 (Lecture Notes in Computer Science)*, Andrew Patrick and Moti Yung (Eds.). Vol. 3570. IFCA, Springer, Berlin, 125–140.

Rima Belguechi, Patrick Lacharme, and Christophe Rosenberger. 2012. Enhancing the privacy of electronic passports. *International Journal of Information Technology and Management* 11, 1/2, 122–137.

Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. 2001. Key-privacy in public-key encryption. In *Advances in Cryptology – ASIACRYPT 2001 (Lecture Notes in Computer Science)*, Colin Boyd (Ed.). Vol. 2248. IACR, Springer, Berlin, 566–582.

Leo Benedictus. 2006. A brief history of the passport. *The Guardian* November 17th, 2006.

BSI. 2009. German Federal Office for Information Security, Technical Guideline TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI), Version 2.01.

BSI. 2010. German Federal Office for Information Security, Technical Guideline TR-03129: PKIs for Machine Readable Travel Documents – Protocols for the Management of Certificates and CRLs, Version 1.10.

BSI. 2013. German Federal Office for Information Security, Technical Guideline TR-03129-2: PKIs for Machine Readable Travel Documents – Protocols for the Management of Certificates and CRLs - National Protocols for ePassport Application, Version 1.1.

BSI. 2015. German Federal Office for Information Security, Technical Guideline TR-03110-1: Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Tokens, Part 1: eMRTDs with BAC/PACEv2 and EACv1, Version 2.20.

Central Intelligence Agency. 2011. Surviving Secondary: An Identity Threat Assessment of Secondary Screening Procedures at International Airports. Available at: https://www.wikileaks.org/cia-travel/secondary-screening/WikiLeaks_CIA_Assessment_on_Surviving_Secondary_Screening.pdf.

Dario Carluccio, Kerstin Lemke, Christof Paar, and Ahmad-Reza Sadeghi. 2006. E-passport: The global traceability or how to feel like a UPS package. In *Workshop on Information Security Applications – WISA'06 (Lecture Notes in Computer Science)*, Jae-Kiwang Lee, Okyeon Yi, and Moti Yung (Eds.). Vol. 4298. Springer, Berlin, 391–404.

Tom Chothia and Vitaliy Smirnov. 2010. A traceability attack against e-passports. In *14th International Conference on Financial Cryptography and Data Security – FC'10 (Lecture Notes in Computer Science)*, Radu Sion (Ed.). Vol. 6052. IFCA, Springer, Berlin, 20–34.

Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. 2009. Physical-layer identification of RFID devices. In *18th USENIX Security Symposium – USENIX'09*. USENIX Association, Montreal, Canada, 199–214.

George Davida and Yvo Desmedt. 1988. Passports and visas versus IDs. In *Advances in Cryptology – EUROCRYPT'88 (Lecture Notes in Computer Science)*, Christoph G. Günther (Ed.). Vol. 330. IACR, Springer, Berlin, 183–188.

Yvo Desmedt. 1995. Securing traceability of ciphertexts – Towards a secure software key escrow system (extended abstract). In *Advances in Cryptology – EUROCRYPT'95 (Lecture Notes in Computer Science)*, Louis C. Guillou and Jean-Jacques Quisquater (Eds.). Vol. 921. IACR, Springer, Berlin, 147–157.

European Commission. 2009. Decision of the European Commission C(2006)2909. Establishing the technical specifications on the standards for security features and biometrics in passports and travel documents issued by Member States.

Klaus Finkenzeller. 2009. Known attacks on RFID systems, possible countermeasures and upcoming standardisation activities. In *5th European Workshop on RFID Systems and Technologies*.

Edgar Friedrich. 2006. The introduction of German electronic passports. In *Second Symposium on ICAO-Standard, MRTDs, Biometrics and Security*.

Frontex. 2011. Operational and Technical security of Electronic Passports. Report by PwC, Collis and the Radboud University for the EU Agency for the Management of Operational Cooperation at the External

Borders (FRONTEX), 2011. https://www.frontex.europa.eu/assets/Publications/Research/Operational_and_Technical_Security_of_Electronic_Pasports.pdf.

Gerhard P. Hancke. 2011. Practical eavesdropping and skimming attacks on high-frequency RFID tokens. *Journal of Computer Security* 19, 2, 259–288.

Jaap-Henk Hoepman, Engelbert Hubbers, Bart Jacobs, Martijn Oostdijk, and Ronny Wichers Schreur. 2006. Crossing borders: Security and privacy issues of the European e-passport. In *Advances in Information and Computer Security, First International Workshop on Security – IWSEC'06 (Lecture Notes in Computer Science)*, Hiroshi Yoshiura, Kouichi Sakurai, Kai Rannenberg, Yuko Murayama, and Shin-ichi Kawamura (Eds.), Vol. 4266. Springer, Berlin, 152–167.

ICAO. 2008a. International Civil Aviation Organization – Machine Readable Travel Documents – Part 3: Machine Readable Official Travel Documents – Volume 1: MRTDs with Machine Readable Data Stored in Optical Character Recognition Format.

ICAO. 2008b. International Civil Aviation Organization – Machine Readable Travel Documents – Part 3: Machine Readable Official Travel Documents – Volume 2: Specifications for Electronically Enabled MRTDs with Biometric Identification Capability.

ICAO. 2010. International Civil Aviation Organization – Technical Report, Supplemental Access Control for Machine Readable Travel Documents.

ICAO. 2013a. International Civil Aviation Organization – MRTD Report: The ICAO PKD State of Play - Future Perspectives.

ICAO. 2013b. International Civil Aviation Organization – Supplement to ICAO Doc 9303 – Release 12.

ISO. 2005. ISO/IEC 7816-4:2013, Identification cards – Integrated circuit(s) cards with contacts – Part 4: Organization, security and commands for interchange.

ISO. 2006. ISO/IEC 7816-3:2013, Identification cards – Integrated circuit(s) cards with contacts – Part 3: Electrical interface and transmission protocols.

ISO. 2008. ISO/IEC 14443-4:2011 Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 4: Transmission protocol.

ISO. 2011. ISO/IEC 14443-3:2011 Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anticollision.

Ari Juels, David Molnar, and David Wagner. 2005. Security and privacy issues in e-passports. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm'05*. IEEE, IEEE Computer Society, Athens, Greece, 74–88.

Burt Kaliski. 2000. PKCS #5: Password-Based Cryptography Specification Version 2.0 – RFC 2898.

Jeremy Kirk. 2007. UK Biometric passports not secure. *PCWorld* March 6, 2007.

Eleni Kosta, Martin Meints, Marit Hansen, and Mark Gasson. 2007. An analysis of security and privacy issues relating to RFID enabled ePassports. In *New Approaches for Security, Privacy and Trust in Complex Environments*, Hein Venter, Mariki Eloff, Les Labuschagne, Jan Eloff, and Rossouw Solms (Eds.). IFIP International Federation for Information Processing, Vol. 232. Springer, 467–472.

Yifei Liu, Timo Kasper, Kerstin Lemke-Rust, and Christof Paar. 2007. E-passport: Cracking basic access control keys with COPACOBANA. In *Special-Purpose Hardware for Attacking Cryptographic Systems – SHARCS'07*. Vienna, Austria.

Marci Meingast, Jennifer King, and Deirdre K. Mulligan. 2007. Embedded RFID and everyday things: A case study of the security and privacy risks of the US e-passport. In *IEEE International Conference on RFID, 2007*. IEEE, 7–14.

Jean Monnerat, Serge Vaudenay, and Martin Vuagnoux. 2007. About machine-readable travel documents. In *Workshop on RFID Security – RFIDSec'07*. Malaga, Spain.

Markus Mösenbacher. 2013. *Preventing fraud in ePassports and eIDs – Security Protocols for Today and Tomorrow*. Technical Report 9397 750 17377. NXP Semiconductors.

NIST/SEMATECH. 2013. NIST/SEMATECH e-Handbook of Statistical Methods. Retrieved December 26, 2015 from http://www.itl.nist.gov/div898/handbook/.

Elisabeth Oswald, Stefan Mangard, and Thomas Popp. 2007. *Power Analysis Attacks – Revealing the Secrets of Smartcards*. Springer-Verlag.

Parliament of England. 1414. Safe Conducts Act.

Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang. 2008. An on-line secure e-passport protocol. In *Information Security Practice and Experience (Lecture Notes in Computer Science)*, Liqun Chen, Yi Mu, and Willy Susilo (Eds.), Vol. 4991. Springer, Berlin, 14–28.

PC/SC Workgroup. 2007. Interoperability Specification for ICCs and Personal Computer Systems - Part 3. Requirements for PC-Connected Interface Devices.

Ivo Pooters. 2008. Keep out of my passport: access control mechanisms in e-passports. http://danishbiometrics.files.wordpress.com/2010/05/ivo.pdf (From https://dl.acm.org/citation.cfm?id=2487439 An investigative analysis of the security weaknesses in the evolution of RFID enabled passport).

Niels Provos and David Mazières. 1999. A future-adaptable password scheme. In *USENIX Annual Technical Conference, FREENIX Track*. Monterey, CA, 81–91.

Amir Rahmati, Mastooreh Salajegheh, Dan Holcomb, Jacob Sorber, Wayne P. Burleson, and Kevin Fu. 2012. TARDIS: Time and remanence decay in SRAM to implement secure protocols on embedded devices without clocks. In *21st USENIX Security Symposium (Security'12)*. USENIX Association, Bellevue, WA, 221–236.

Henning Richter, Wojciech Mostowski, and Erik Poll. 2008. Fingerprinting passports. In *NLUUG Spring Conference on Security*. 21–30.

Luigi Sportiello. 2012. Weakening epassports through bad implementations. In *Workshop on RFID Security – RFIDSec'12 (Lecture Notes in Computer Science)*, Jaap-Henk Hoepman and Ingrid Verbauwhede (Eds.), Vol. 7739. Springer, Berlin, 123–136.

Eileen Sullivan. 2009. Post-9/11 reforms don't stop passport fakery. *The Associated Press* March 13, 2009.

Alexandra Topping. 2008. Thousands of blank passports stolen in raid could be used in fraud or terrorism. *The Guardian* July 30, 2008.

Received November 2013; revised June 2015; accepted September 2015