P. Najera, F. Moyano, and J. Lopez, "Security Mechanisms and Access Control Infrastructure for e-Passports and General Purpose e-Documents", *Journal of Universal Computer Science*, vol. 15, pp. 970-991, 2009.
http://doi.org/http://dx.doi.org/10.3217/jucs-015-05-0970
NICS Lab. Publications: https://www.nics.uma.es/publications

Security mechanisms and access control infrastructure for e-Passports and general purpose e-documents

Pablo Najera
(University of Malaga, Malaga, Spain
najera@lcc.uma.es)

Francisco Moyano
(University of Malaga, Malaga, Spain
moyano@lcc.uma.es)

Javier Lopez
(University of Malaga, Malaga, Spain
jlm@lcc.uma.es)

Abstract: Traditional paper documents are not likely to disappear in the near future as they are present everywhere in daily life, however, paper-based documentation lacks the link with the digital world for agile and automated processing. At the same time it is prone to cloning, alteration and counterfeiting attacks. E-passport defined by ICAO and implemented in 45 countries is the most relevant case of hybrid documentation (i.e. paper format with electronic capabilities) to date, but, as the advantages of hybrid documentation are recognized more and more will undoubtedly appear. In this paper, we present the concept and security requirements of general-use e-documents, analyze the most comprehensive security solution (i.e. ePassport security mechanisms) and its suitability for general-purpose e-documentation. Finally, we propose alternatives for the weakest and less suitable protocol from ePassports: the BAC (Basic Access Control). In particular, an appropriate key management infrastructure for access control to document memory is discussed in conjunction with a prototype implementation.

Keywords: electronic documents, e-documents, e-Passport, RFID technology, RFID security, security mechanisms, access control infrastructure, pervasive computing, security
Categories: D2.1, D2.11,K6.5

1 Introduction

Despite the spread and integration of computers in almost every facet of our daily life and the transition from physical documents to purely digital-based information in a wide range of scenarios, paper-based documentation is not likely to disappear. Traditional methods of recording and transferring data do not suffer from technical reliability problems common in the digital information world (e.g. a system failure or infrastructure connection problem), nor do they require special hardware or equipment and they can be used by individuals without special technological training or preparation.

However, the processing of traditional paper-based documentation inevitably requires physical handling: the content must be read, interpreted and, if required, the relevant data must be input into the information system by hand. As a result, traditional documentation processing is a slow and error prone task. Furthermore, most documents lack adequate security mechanisms and rely on handwritten signatures or stamps which can be easily forged. As a result, the issuer of a document can not be properly authenticated and it may be difficult to adequately prevent content alteration and counterfeiting. These security problems have been extensively addressed in electronic documents using techniques such as digital certificates and document signature to enhance the security properties.

In our view, paper-based documentation should be integrated with the information system obtaining automatic processing capabilities and enabling the use of cryptographic security mechanisms in traditional documentation. This link between the physical and digital world can be obtained by means of the RFID technology.

In a nutshell, RFID (Radio Frequency IDentification) technology [Lahiri 2005] enables wireless data transmission by means of a miniature integrated circuit equipped with an antenna that can be attached to or embedded in the object to be controlled. This tag or transponder is interrogated by a reader in order to obtain the data stored in its memory, which is typically referred to the related item. Collected data is appropriately processed by an information system and information on the tag is updated through the remote reader if required. By integrating an RFID tag into paper-based documentation, it can benefit from the advantages of digital information processing without sacrificing the reliability and convenience provided by the physical support.

The implementation of RFID technology focuses mainly on tags as a new generation barcode due to non-contact reading capabilities without requiring direct line of sight while providing a unique identification code for each singular item (as opposed to a generic class or type code provided by barcodes). For this reason, it is being widely adopted in the retail, transportation, military and pharmaceutical industries. At the same time, security research on RFID has focused on the prevention of ID code disclosure in order to protect user privacy and to avoid concealed object identification and individual tracking practices [Najera 2007].

However, apart from seamless identification capabilities, we foresee that features such as pervasive computation and memory storage have also the potential to transform other areas, in particular personal and legal documentation by providing a transparent link to the information system and advanced security features. A pioneer in this approach and, to the best of our knowledge, the most representative example of documentation taking this hybrid approach to-date is the ICAO electronic passport currently in use in 45 countries around the world. The traditional passport has been enhanced using RFID technology to speed up traveller recognition by border security officers and to improve passport authenticity validation. As the advantages of hybrid documentation gradually become apparent, other personal e-documents will potentially appear.

In this paper, we introduce our general use e-document concept, analyze the weaknesses in existing security mechanisms defined for e-Passports and their suitability for other kinds of electronic documentation. The BAC (Basic Access Control) security mechanism as defined in the ICAO standard [ICAO 2005] is shown

to be the most vulnerable mechanism and least suitable for general documentation. Suitable alternatives for the actual definition of BAC are discussed. In particular, a key management infrastructure for access control in electronic documentation is presented with a working prototype implementation.

In section 2, the ICAO e-Passport is introduced, the reasons that motivated the choice of RFID technology for this particular e-document (which are also relevant for a general e-document approach) and an explanation of the security mechanisms defined in the ICAO Doc 9303 standard. The Basic Access Control (BAC) mechanism is explained in more detail as it will be the centre of later discussion and proposals.

In section 3, our vision of a general use e-document is outlined. We describe how paper-based documentation can benefit from the 'hybrid approach', the new requirements that can be requested from a document because of the natural link between it and the information system, as well as the security requirements an e-document can fulfil. Some examples of general use e-documents are presented.

Section 4 analyses the weaknesses that have been presented in the literature for the ICAO security mechanisms, as well as the suitability of these security mechanisms for general use e-documents. Most of the issues are derived from the key derivation scheme for the Basic Access Control mechanism which is closely linked to the e-Passport data fields.

Different alternatives for this key generation scheme which focus on solving the low key entropy problem are outlined in Section 5. These approaches enable the use of a dynamic access key and provide a generic scheme not correlated with e-document internal data fields.

Section 6 goes into a previous alternative in depth and proposes a key management infrastructure for handling control access keys for e-documents which do not suffer from the problems presented in section 4 and a prototype infrastructure implementation is shown. Finally, in section 7 some conclusions are outlined.

2 Security mechanisms in the ICAO e-Passport

2.1 The ICAO e-Passport and the selection of RFID technology

As the culmination of several years of work, which began in 1998, the International Civil Aviation Organisation (ICAO) and, in particular, the New Technologies Working Group (NTWG) that belongs to the Technical Advisory Group on Machine Readable Travel Documents (TAG/MRTD) developed the e-passport guideline. This standard focuses on a biometric enabled passport as the result of studying how identity confirmation with travel documents can be potentially enhanced by means of biometrics for strengthening border security.

Several factors were taken into consideration by ICAO for selecting the most suitable technology for the development of electronic travel documents. Firstly, the technology should be non proprietary and should provide global interoperability. In addition, it should not involve a considerably larger amount of work to use (e.g. it may be necessary to place the document in a reading device in a specific position) while not requiring changes of traditional format from prior versions of the document

(i.e. usable in paper-made book-style documents) and provide enough memory to store appropriate data (e.g. biometric details such as a facial image). These guidelines for technology selection, which are not specific only for travel documents, but shared by other types of documentation as proposed in this paper, led ICAO to select RFID technology. According to [Jacobs 2006], further reasons for selecting "proximity" smartcards instead of contact smartcards involves also less wear through document lifetime and higher data rates.

The ICAO DOC 9303 standard for MRTD proposes the use of RFID-based contactless smartcards conforming to the ISO-14443 standard [ISO 2000], embedded in the document booklet. The microchip mirrors the same personal data that is printed on the document and the MRZ (Machine Readable Zone) and has a copy of the facial image and optional secondary biometrics (i.e. fingerprint and iris image).

2.2 Security mechanisms in the first and second e-Passport generations

As a starting point on the discussion of appropriate security mechanisms for personal electronic documents, we will first present the security features already defined by ICAO for the e-passport with a subsequent analysis of their weaknesses and suitability in our extended e-document concept.

Security features in the first and actual generation of e-Passports are based on three security protocols:

- *Basic Access Control* (BAC): an optional security mechanism oriented to provide a basic mutual authentication as well as an encrypted communication channel between the inspection system and RFID chip. This mechanism is used in both the existing e-Passport and the EU proposed e-passports with extended security. Due to this security feature being a critical pillar in our later discussion on weaknesses in e-passports, suitability for other types of e-documents and our alternatives, a more detailed explanation of this security mechanism will be presented in [Section 2.3]
- *Passive Authentication* (PA): a mandatory security mechanism that allows an inspection system to verify the integrity and authenticity of the data stored in the chip. A hash (i.e. digest version) of each data group is also stored in the document signed by the issuing state using the Document Signer private key. By verifying this signature, the inspection system verifies whoever is certifying the information in the e-document and the integrity of the document.
- *Active Authentication* (AA): an optional ICAO security feature designed to prevent chip modification or cloning (i.e. duplication of the readable areas of the chip, namely the Logical Data Structure and the Document Security Object SOd areas of the chip). In order to authenticate the tag, the microchip contains an active authentication key pair: the public key is stored in the SOd while the private key is kept in secure memory. The challenge-response protocol used to verify that the chip is not a copy complies with the ISO/IEC 7816 Internal Authenticate mechanism following this scheme: the reader generates and sends a random nonce to the document. The tag digitally signs the challenge using its private key and sends the answer to the reader. By means of the public key obtained from the SOd, the inspection system verifies the signature. The Active Authentication mechanism has been

implemented by a small number of countries [Matyáš 2007] and will be replaced in the second generation of ePassports by the Chip Authentication mechanism.

By 28th June 2009, a second generation of electronic passports will be issued conforming to the European Union proposal. This new generation will require an Extended Access Control (EAC) mechanism in order to access sensitive biometric information (i.e. e-Passport holder's fingerprints) stored in the chip. Security mechanisms in this version, as described in the EU proposal, involve the following protocols:

- *Basic Access Control (BAC)*: this mutual authentication protocol between document and inspection system is preserved from first generation e-Passports.
- *Chip Authentication (CA)*: a mandatory security mechanism that substitutes the Active Authentication mechanism from the previous version MRTD documents. It provides authentication of the chip preventing cloning of e-passports and generates a session key.
- *Passive Authentication (PA)*: as described for first generation ePassports.
- *Terminal Authentication (TA)*: this mandatory EU EAC mechanism allows a chip to verify that an inspection system (IS) has authorization to read the secondary biometrics on the chip. The terminal has to provide a chain of certificates to prove that it is certified by a visiting country's Document Verifier which is in turn certified by the certification authority of the home country. Once the certificate chain has been verified by the e-document, a two-pass challenge response protocol is used to authenticate the inspection system. In the challenge-response protocol, the chip sends a challenge to the IS. The IS signs the challenge and sends the signed challenge back to the chip. If verification of the signature is successful, the IS is authenticated.

2.3 Basic Access Control mechanism

In order to prevent skimming and eavesdropping, ICAO proposed the optional Basic Access Control security mechanism. To protect their privacy, the electronic document's owner should be able to prevent unknown or ill-intentioned readers from accessing the document, while giving his/her consents to a trusted reader. For this purpose, the data present in a Machine Readable Zone (MRZ) printed in the physical document is used to derive the keys required to access the document. In the original mechanism concept, the reader would only have access to the data in the MRZ, and, would therefore been able to derive the keys if the document has been physically opened and presented to the reader by its owner, therefore obtaining the user consent

The following steps take place in a mutual authentication based on the BAC mechanism:

- Border security control obtains suitable data from the MRZ zone by means of an OCR-B scanner or it is typed in. The key derivation mechanism is performed based on this data (e.g. e-passport number, e-passport holder's date of birth, e-passport expiry date and check digits).

- A three-pass challenge response protocol takes place for mutual authentication between the e-passport and the inspection system.
- If authentication is successful, session keys are computed and all further communication is protected using secure messaging.

The key derivation mechanism is used twice: first, to establish the Basic Access Keys (K_{ENC} and K_{MAC}) and later for the session keys used in secure messaging. In order to derive two 3DES keys from a single key seed K_{seed} (formed based on MRZ data) a 32 bit counter c is used updating its value (c=1 for encryption key K_{ENC} and c=2 for MAC computation key K_{MAC}).

Steps performed in the key derivation process are the following:

- $D = K_{seed} \| c$
- $H = SHA\text{-}1(D)$
- K_1= bytes 1..8 of H, K_2= bytes 9..16 of H
- Parity bits of K_1 and K_2 are adjusted to form correct DES keys.

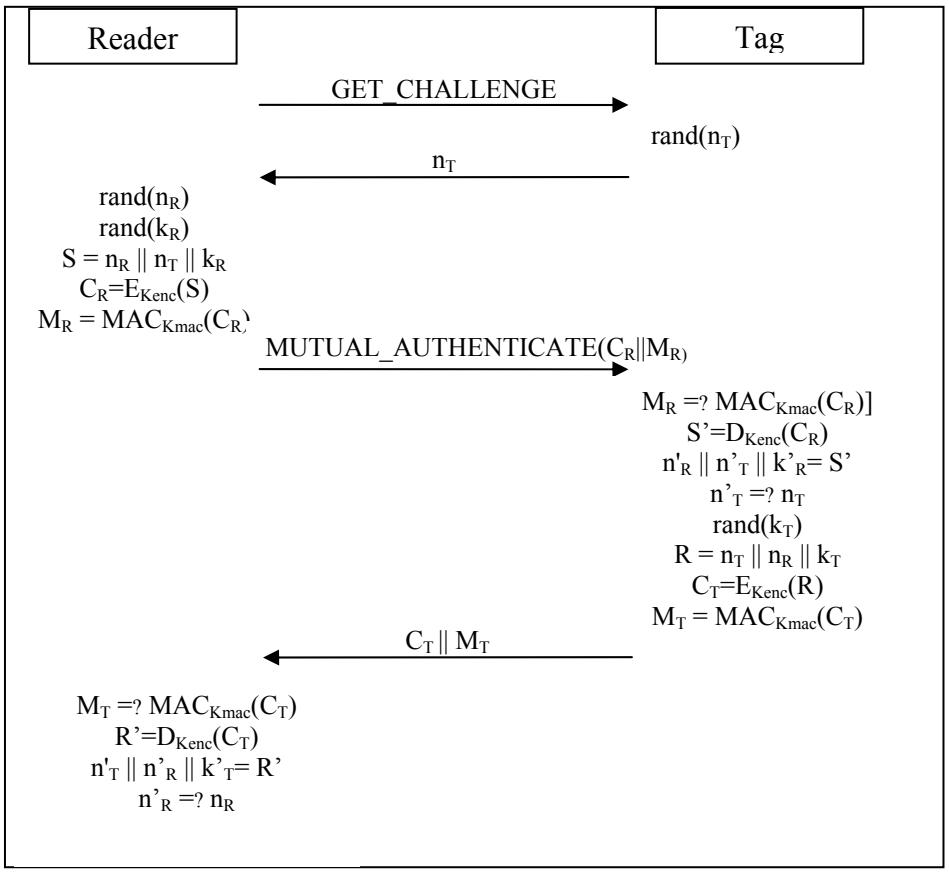

Figure 1: BAC three-pass challenge-response protocol for mutual authentication

The three pass challenge-response protocol conforms to ISO/IEC 11770-2 Key Establishment Mechanism 6. The protocol is initiated by a challenge request from the reader which is answered with a 64 bit nonce generated by the RFID tag. On reception, the reader concatenates its own nonce, the tag's nonce and a reader's generated 128 bit keying material. The ciphertext of this data and its checksum (according to ISO/IEC 9797-1 MAC Algorithm 3) are computed and sent to the tag using K_{ENC} and K_{MAC} as encryption and MAC keys respectively. The tag verifies the checksum, then extracts and checks its nonce from the ciphertext. If verification is successful, the tag computes its own ciphertext based on the concatenation of both nonces and its own generated keying material. The ciphertext and its checksum is sent to the reader. Finally, the reader verifies the validity of the checksum, extracts its nonce from the ciphertext and checks it. If verified, reader and tag are considered mutually authenticated. Both keying materials form a second K_{seed} for the key derivation algorithm, for deriving a session key to encrypt further communication ('secure messaging').

3 From the e-Passport to general use e-documents: hybrid document requirements

Although the e-Passport has been a pioneer and, to the best of our knowledge, the most widely adopted form of e-document that embeds RFID technology into the old-fashioned paper-based document in order to enhance its security and provide seamless integration with the information system, other types of documents can benefit from this approach.

Paper-based documentation may be more convenient and easier to use in daily life scenarios than purely digital documents. Document holders need not be familiar with electronic files, digital certificate management or technology-based procedures (e.g. secure web-based platforms or even simple e-mails) in order to handle documents and use them. Moreover, physical documents are tangible and reliable in the case of a system failure or network error. As a consequence, paper-based documentation is unlikely to disappear completely in favour of pure digital documents. However, these traditional-format documents are slower to process, prone to human error handling, and lack advanced security features to verify document integrity or authenticity.

In our view, not only e-Passports, but a wide range of document types could benefit from a hybrid approach: traditional paper-based documentation could be enhanced with electronic capabilities by means of RFID technology. Nevertheless, a case by case study of each document type would be necessary to define the particular requirements of each application scenario and analyze how they can benefit from the hybrid approach. A terminal with RFID reading/writing capabilities (and an optional network connection depending on the infrastructure architecture) would be required at document issue and checking points. Therefore, non-electronic (e.g. manually written) documents which do not need to be processed by an information system during their lifetime and without technical facilities at issuing or reading points would be out of scope.

Hybrid documentation has the potential to provide a wide set of features which were not even considered in the traditional approach. Novel features that may be required in a hybrid document by a particular scenario focus on two main aspects: seamless integration of documents in the information system and security capabilities. Requirements in seamless integration adequate for hybrid documentation are:

- *Transparent document identification*: the particular instance of the document that is being processed by a terminal needs to be uniquely identified in order to proceed with the task. A unique code (e.g. document number) is required to identify the document. The RFID tag embedded in the document is necessary to provide this ID. Based on this code, additional data related to the document can be retrieved directly from a trusted centralized database and used as required by the application. For example, an invoice could be shown at the point of sale in order to have a product which is still under warranty repaired. After automatic recognition of the invoice, details on the product warranty period or manufacturer could be recovered or the product location and status could be updated.
- *Document content retrieval*: a digital version of the document content is required by the information system for processing tasks. On paper-based documents, human interpretation of document details along with manual input into the system would be required. In some cases, optical character recognition (OCR) techniques could be applied imposing restrictions on document format and layout complexity to maximize OCR accuracy. In hybrid documentation, a digital version of the document properly formatted for automatic processing could be stored directly within the RFID tag. A terminal can directly recover this digital version if there is no central database, if the document had not been previously processed by the system and if its content was unknown by the central server or if a terminal has no connection to the document processing infrastructure. For example, a printed version of a company statistics report on their last quarter wholesales could embed a digital version within an RFID tag attached to the first page of the report to enhance document content access and processing (e.g. advanced analysis of the statistics presented in the document by filtering of relevant data, data-mining contents or comparison with previous statistics reports).

An application scenario may also impose requirements on hybrid documentation in terms of security features that the e-document should comply with. These requirements will also be converted into security goals that mechanisms and protocols implemented in the RFID tag should fulfil:

- *Document source/pedigree*: the inspection point during the verification process requires knowing the origin of the document, who issued it and who certified its content. In paper-based documentation, a textual description of the source (e.g. individual or company name) in conjunction with a handwritten signature or stamp is typically used to certify document origin. In hybrid documentation, a more complete description of the issuer (e.g. a X.509 certificate including its name, a unique identifier and public key) can be stored within the tag. Digital proof of the issuer such as a digital signature of its description and the current document number can be included to

authenticate the issuer. If a digital signature was not bound to the particular document instance, a tamper-proof tag embedding mechanism would be necessary to prevent this tag being used in a different document. A challenge-response protocol for authenticating the chip to thwart document cloning may also be required. As an example of an application scenario which would impose this requirement, the authenticity of a sick note presented by a worker should be appropriately verified by its company to stop home-made forms being forged during intensive load periods.

- *Document integrity validation*: the inspection system requires verifying that the document content has not been altered or modified by any third party since being issued by a trusted issuer. In hybrid documents, the issuer can digitally sign the hash of the on-memory document version. If the digital version is modified, the inspection system will fail during the verification of the signed hash against the actual output of the hash function, therefore detecting the manipulation. However, security mechanisms based exclusively on digital signatures only certify the integrity of the digital document version. If integrity validation of the printed document is required, a comparison between physical and digital versions needs to be done manually (e.g. visual comparison between on-paper and on-screen versions) or automatically (e.g. an OCR-equipped terminal digitalizes the document and checks it against the signed memory content). As an example, a bank would require to verify that the details on a payslip (e.g. document holder's name, gross and net pay amounts) shown by an applicant have not been altered in any way before approving a mortgage application in order to avoid future unpaid instalments.
- *Non Repudiation*: the inspection system requires collecting an undeniable proof during the document processing procedure in order to be able to prove at a later stage that a particular document instance was shown to the terminal and inspected. As an example, a security officer at the entrance to a highly restricted area may need to collect evidence of each company identification pass shown by workers or visitors for later supervision of access events.

These security requirements for hybrid documents are also translated into requirements for the security mechanisms. Additional security goals [Vijayakrishnan 2008] that mechanisms should comply with are:

- *Non ID disclosure*: in order to protect the document holder's privacy, an unauthorized reader should not be able to obtain any data from the RFID chip that provides information about the particular document instance under scrutiny. In particular, the RFID tag should not provide a static unique code during the tag singulation process which takes place each time an IS-RFID tag communication is started. If a static number were to be used, an attacker could use this code to silently identify the e-document or track its owner. As a better alternative, a random number for tag singulation should be used in each session. In this approach, the e-document would only provide identifiable document details after the inspection system has been properly authenticated. If the inspection system requires any document specific details

in order to complete authentication these details would be obtained from a different information source (e.g. the MRZ document zone).

- *Data confidentiality*: document details should remain secret and only known by authorized parties. Therefore, access to data in the document chip should require prior authentication of the requester. The communication channel between the document chip and the inspection system should be secured (e.g. by means of data encryption).
- *Mutual Authentication*: document chip and inspection system should be mutually authenticated before document details are transmitted or updated.
- *Key freshness and integrity*: nonces used in protocols must not have been used in any previous execution of the protocol. Moreover, both the document chip and inspection system should be able to verify: i) the session key integrity: derived session key has not been altered ii) the other party has derived the same session key.
- *Forward secrecy*: no future communication should be compromised by the loss of a session key or the key used to generate it.

Depending on the specific application scenario of each hybrid document type, a different subset of the previous requirements will be considered. In the Machine Readable Travel Document (MRTD) approach, all of these requirements are applicable and have been addressed in one way or another in the security mechanisms for the electronic passport presented in [Section 2.2]. The transparent document integrity validation requirement imposes restrictions on the physical characteristics of the paper-based document and therefore would limit the range of documents where it could be fulfilled. In the MRTD solution, it has been solved by means of the inclusion of Machine Readable Zone (MRZ), a "zone" in the document which mirrors document details in a easily machine readable format. In a more general approach, the document itself and its layout could be predefined to facilitate interpretation of printed data by an OCR-capable electronic scanning device (i.e. using an easy-to-read font type with specific size and predetermined data elements and its relative position in the document layout). In any case, this approach limits the automatic integrity validation of printed data to documents with a restricted amount of textual information (e.g. in the e-Passport case, the printed facial image is not validated by the scanning device).

These security requirements and restrictions would be appropriate for a wide range of typically paper-based documents which contain a limited amount of data issued by a trusted authority, but where document authenticity and integrity should be verified. These documents include: certificates of personal life events (e.g. death/birth certificate, marriage, divorce or adoption records), holder's curriculum vitae (e.g. a university degree or academic qualification certificate), medical documents (e.g. sick note or medical prescriptions) and monetary documents (e.g. payslips, wire-transfer slips or invoices).

This hybrid document approach would also be appropriate for personal identification documents developed for commercial purposes or for internal use in organizations. Thus, it would not be required the use of official personal identification documents such as e-passports in non-governmental, but commercial applications which otherwise would have access to the sensitive personal data stored in the official

personal document and could threat holder's privacy. Ad-hoc application specific identification documents would avoid third parties with a professional or commercial relationship with the holder accessing the personal and biometric data available in e-passports and therefore, the risk of document forgery would be avoided. Moreover, access to the ICAO PKD is restricted to authorized readers, and as a result verification of e-Passports authenticity by third party applications would not be feasible.

If some security goals are not required by a particular hybrid document type, a subset of the security requirements could be dispensed with (e.g. if document authenticity is not necessary, the chip authentication mechanism would not be required).

Although security mechanisms defined in the ICAO standard focus on MRTD, to the best of our knowledge, they form the most comprehensive and suitable suite of mechanisms for RFID documents. In the next section, we will analyze security mechanisms in e-Passports taking into consideration both their weaknesses and suitability for general personal documentation according to our concept of personal documentation and the security requirements mentioned previously.

4 Weaknesses of e-Passport security mechanisms and suitability for general use e-documents

As described in [Section 2.2], the first security mechanism that takes place in both existing e-Passports and the proposed EAC EU e-Passports is the Basic Access Control mechanism. BAC provides a basic mutual authentication and an encrypted communication channel between the IS and e-document to prevent skimming and eavesdropping. However, it has been pointed out that BAC has a number of important weaknesses [Juels 2005] namely low key entropy and a fixed access key.

According to the recommendation of the National Institute of Standards and Technology (NIST) [NIST 2007] the minimum key length for general purpose protection was 80 bits in 2008. The ECRYPT EU Network of Excellence on cryptology recommended a key length of 112 bits taking into consideration the evolution in computation in twenty years [ECRYPT 2008]. The access key seed K_{seed} used in BAC for key derivation has a length of 128 bits which would exceed previous recommendations providing an adequate security level. However, as shown in [Section 2.3], K_{seed} is derived from the MRZ information, namely the passport number and expiry date, holder's date of birth and the check digits of each field. As a consequence, the entropy of K_{seed} depends on the entropy of these fields.

Theoretically, the entropy of the date of birth field has been estimated as 15.15 bits [Avoine 2008] (14 bits in [Juels 2005]), date of expiry as 11.83 bits [Avoine 2008] (11 bits in [Juels 2005]) for an ePassport validity period of 10 years, while the entropy of the passport number depends on the numbering scheme of each country with a maximum of 46.53 bits [Avoine 2008]. Therefore, the maximum theoretical key entropy is estimated as 73 bits, although dependency on the implementation of the issuing nation may reduce the real entropy of the key. The ICAO PKI Technical Report acknowledges that key entropy in BAC is at most 56 bits, but even some of these bits may be guessable under specific circumstances.

In the case of US passports, 2 digits of the passport number are dedicated to the encoding of 15 issuing offices, while 7 digits are arbitrary, limiting the entropy of the passport number to 27 bits and the total key entropy to 52 bits. In the German ePassport, only numeric characters are used for the passport number and its format is structured (4 digits represent the BKZ local issuing agency and 5 digits the serial number) providing 30 bits of entropy of this field and a total entropy of 55 bits. If a rough demographic model is known to the attacker, the entropy of passport number is reduced to 26 bits [Carluccio 2006]. Using further assumptions (i.e. the attacker knows a pair of passport numbers and its corresponding expiry date that belongs to that BKZ, as well as the city of residence and birth date of the victim) key entropy may be reduced to below 20 bits. In the case of Belgian ePassports [Avoine 2008], a 2 letter prefix in the passport number and validity period of 5 years leaves 54 bits of overall entropy. However, passport numbers are assigned sequentially so a strong correlation between this number and the issue date exists. Taking into consideration other factors such as the fact that ePassports are issued roughly 250 days a year, blocks of numbers are assigned to each official language (i.e. Dutch, French or German) and the starting ePassport issuing date is known, the result is an effective key entropy of 38 bits that can be reduced to 23 bits if the date of birth is known.

Brute forcing the BAC key can be done by means of an on-line attack (testing each key against the IC chip) or off-line attack (i.e. first eavesdropping the communication between an RFID tag and legitimate reader and later testing each key against the capture frames). The low effective entropy of the BAC key makes an on-line attack on an ePassport feasible if heuristics are applicable (e.g. in [Avoine 2008], Avoine et. al were able to “break into” a Belgian ePassport in a matter of minutes), while a 23 bit entropy can be “broken into” in a second with a conventional PC.

Although implementation issues, as exposed in the previous study cases, substantially reduce the entropy of BAC access keys, even a completely random numbering scheme and optimal implementation would not exceed the theoretical maximum of 73 bits which falls below the recommended key sizes. Therefore, a different key derivation procedure for the BAC protocol that provides higher key entropy should be used in order to comply with recommended minimum security levels.

An additional problem for the ICAO BAC mechanism comes from the fact that the key is derived directly from the static MRZ information, and is therefore a fixed key [Juels 2005]. Consequently, once an inspection system at a border office (or a commercial third party such as a hotel or a bank) has had access to the information in the MRZ zone, the BAC key is known for the lifetime of the document. Access to the document data protected under BAC can not be revoked. Moreover, sharing BAC keys with an unauthorized third party (which may never have come into contact with the document and read the MRZ data before) would provide it with irrevocable access to the ePassport.

Last, but not least, regarding the weaknesses of the BAC mechanism and its suitability for general approach e-documents, the use of specific data fields associated with a passport document (i.e. passport number, expiry date and holder's date of birth) for key derivation is document dependent, and, therefore, is not reusable in other types of hybrid documents. A generic document independent K_{seed} derivation process would be required for BAC use in general purpose e-documents.

Analyzing the heart of the matter, all the weaknesses of the BAC mechanism and its unsuitability for e-documents are due to the same part of BAC: the key derivation process up to obtaining K_{seed} (following the procedure shown in [Section 2.3]). Security issues are caused by the use of data fields for key generation which are specific to a passport, with a limited range of values, static and guessable for an attacker with basic information about the victim (e.g. physical appearance of the holder provides clues about his/her age and origin, therefore birth date and applicable heuristics for his/her passport). If a secure K_{seed} derivation method were adopted, these issues would be resolved. The access control mechanism itself is adequate and its implementation is still recommended for MRTD documents [Vijayakrishnan 2008].

In [Section 5], we will discuss alternatives to the actual BAC specification in order to increase key entropy, provide a dynamic access key and generalize the solution for other types of hybrid documents. In particular, in [Section 6] we will discus a key management infrastructure which would solve all the issues presented. A prototype implementation of this idea is shown in [Section 6.3].

Following our analysis of the weaknesses of ICAO security mechanisms and their suitability for hybrid documents, to the best of our knowledge, no security fault has been found in the literature for the Passive Authentication mechanism incorporated into both existing ePassports and the EU EAC proposal. Regarding its suitability for generic hybrid documents, although ePassports integrate high-end RFID chips which feature 3DES, RSA, Diffie-Hellman key agreement and Public Key cryptography, lower-end RFID chips may be used in less security-aware document types. Even though Passive Authentication is based on public key cryptography, the IC chip is only required to store the signed data and certificates. Computation necessary to verify the integrity and authenticity of digitally signed data hashes is performed by the inspection system, not by the RFID chip itself. Therefore the PA scheme is considered secure and appropriate for hybrid documentation.

On the other hand, the Active Authentication mechanism provides a chip authentication method based on public key cryptography, therefore, documents that would need to implement this mechanism would require higher cost IC chips. In [Vijayakrishnan 2008] a formal verification of security mechanisms in actual e-Passports is done. Authors conclude that, assuming BAC protocol were secure, Active Authentication itself presents no security weakness. However, if an attacker is able to successfully execute a brute force attack and recover the BAC seed key K_{seed}, the following security goals are not fulfilled:

- *Mutual authentication*: once the attacker is in possession of the initial keys, he can authenticate to the document impersonating an authorised reader. The attacker can also recover K_{seed} used in the AA protocol and authenticate to the reader as a genuine document.
- *Data confidentiality*: the attacker is able to compute K_{ENC} and, therefore, decrypt any communication of the document with an authorised reader.
- *Key freshness*: nonces may not be fresh and may not be generated with a good pseudo-random algorithm.
- *Key integrity*: the attacker is able to decrypt communication and encrypt his own messages; therefore key integrity can not be assured.

- *Forward secrecy*: once K_{seed} has been recovered by the attacker, any future communication (during the document validity period) will be at risk.

Once again, it is required that the BAC protocol be carried out in a secure way. As explained before, sections [Sections 5,6] will provide solutions to these aspects.

The EU EAC proposal introduces two new security mechanisms (i.e. Chip Authentication and Terminal Authentication) as explained in [Section 2.2]. These mechanisms require public key cryptography capable IC chips in documents and impose strong memory and computation requirements on the RFID tag.

A complex certificate and key management infrastructure is required which represents an important challenge from an organizational point of view [Matyáš 2007] due to cross certification requirements between countries (e.g. each country needs to certificate the document verifiers from other countries) and management of certificate revocation lists. If the scheme of these mechanisms were adopted in other types of hybrid documents, a similar certificate management problem would need to be solved. Anyway, depending on the extension of the document type, the number of authorised document issuers and inspection systems, the dimension of the problem (i.e. complexity of the certificate management infrastructure) could be substantially reduced.

Moreover, the document IC chip needs to be able to handle and verify a complete chain of certificates from the IS certificate, right from a visiting country's Document Verifier certificate to the home country's Certification Authority. Even high-end RFID chips will have limited memory and computing capabilities, so an attacker could perform a DOS attack flooding the chip with a large number of certificates [Vijayakrishnan 2008].

Last, but not least, even though certificates have an expiry date, passive RFID chips have no time control (i.e. they lack an internal clock) so an inspection system could use an expired certificate to be authenticated to a document IC chip and access its data. The document could update its memory with the certificate from the IS each time it is accessed to avoid the used of certificates whose expiration date is older than the date the last certificate was issued, but this would only solve the problem partially as it would depend on the certificate issue date.

In the key management infrastructure presented in [Section 6], it is possible to update the BAC access key each time an e-document is accessed. This approach of updating the access key instead of a complete certificate would require less time and memory. Furthermore, if the BAC protocol is turned into a mandatory protocol before the Active Authentication takes place, this approach could prevent the attack based on expired certificates. This is due to the fact that once the document access key has been updated, no IS with an outdated key would be able to access the document. In order to obtain the actual key, the IS is required to authenticate against the key server using its certificate. The key server does not lack an internal clock and it is, therefore, not vulnerable to the outdated certificate attack. As a result, if the IS is able to recover the actual access key from the key server, it means that the status of its certificate has been validated.

5 Alternatives for a secure access key

As we have shown before, the actual key derivation mechanism for access control defined in BAC has some important security issues. In this section, several alternatives are discussed which increase the low key entropy of the actual BAC scheme, provide a dynamic access key and generalize the solution for other types of hybrid documents.

The low key entropy problem is due to the fact that the key has not a random origin, but rather it has been derived from specific document fields with a limited range of values. In a first approach, the access key value could be separated from information related to the document or holder, thus limiting the amount of data that an attacker could guess. The key could be securely generated from random data and printed directly in the MRZ zone by the document issuer. Using this approach, key entropy could be increased up to the maximum value (e.g. 128 bits if K_{seed} is directly obtained from this value). However, this approach has two important drawbacks: first, if an attacker is able to obtain physical access to the document, he would be able to read the key and, therefore, access tag memory content and perform further attacks based on this information (e.g. cloning the e-document). Secondly, printing this key value over the MRZ implies a static key value during the document lifetime, so access to the e-document would not be revocable.

The key value could also be isolated from the information in the MRZ document zone. A key ID instead of the key value itself could be printed in the MRZ. This key ID would be required to recover the key value available in a different information source (e.g. an access key server). As information in the MRZ is static, this approach would enable the use of a dynamic access key. Moreover, once physical access to the document is obtained, all the information in the MRZ is known. Separating the key value from the data in the MRZ, additional control on who recovers the key value can be established. On requesting a key value from a key ID, the inspection system could be required to authenticate the access key server. As the authentication between the IS and key server does not suffer from the computational constraints of IS-document communication, a higher security level (e.g. certificate-based authentication) can be used in this process. On the other hand, a connection between the IS and the key server would be required to recover the access key, which may limit the range of e-documents where this approach can be adopted. A local key cache could be used to minimize this problem.

In the case where the document lacks an MRZ zone, the key ID could be stored in the memory of the tag. However, a static number in the RFID tag would enable an attacker to read this identification at a distance without the knowledge of the document holder, thus raising tracking and privacy issues [Najera 2007].

An alternative would consist of storing the key identifier inside an encrypted file in a non-restricted chip memory area, while the document information would be in an access restricted area. So, in order to obtain a key, first it would be necessary to decrypt the file, and then, ask the server for the access token. However, we face a new key management problem to store and retrieve the key to decode the file. If a simple scheme with a fixed unique encryption key is considered, anyone who guesses this key would gain access to all the e-documents sharing it. On the other hand, if a different and dynamic key was used for encrypting the file, a key management system

would be required, thus coming back to our original problem. Also, a static encrypted key would also face the tracking problem.

Finally, if the document can be optically recognized, but no additional data should be physically printed in the document (e.g. the MRZ zone should not be modified) a combination of the original BAC scheme and key server approach could be used (applying the original BAC scheme to grant access to the key ID). In this case, two protected areas would be used inside the memory: one of them would be for storing the key identifier and would be restricted to perform the BAC before it was read. The other one would store the e-document data. So, this solution adds an additional security level to the actual BAC implementation before enabling access to document details: after performing BAC, the inspection system is required to authenticate the key server to recover the access key. In each e-document type, specific data elements available in the document should be selected for the initial K_{seed} generation. Although the security level would be substantially increased, the authentication process would be slower, since it requires two phases, and the access to the key ID would share the same limitations as the actual BAC.

Considering this analysis, we will focus on the second approach: e-documents which integrate an MRZ zone. MRZ data will contain the key ID, but will not be correlated with the access key value stored in a secure server. In the next section, we further develop the requirements and design of the access key management infrastructure.

6 Access key management infrastructure for e-documents

Now, we are going to describe several aspects of our proposed key management infrastructure. First, we will discuss the architecture of the infrastructure, identifying all the components and how they are related to each other. Secondly, we will explain the security requirements which are needed in order to make our infrastructure safe against eavesdropping and to protect it from data leakage. Finally, we will describe a prototype implementation of our e-document infrastructure based on open-source components.

6.1 Infrastructure Architecture

In a birds-eye view, we can characterize our infrastructure as three different modules working together and cooperating with each other, the client module being deployed in the inspection system, the server module and the secure key storage module embedded in the latter, as shown in Figure 2.

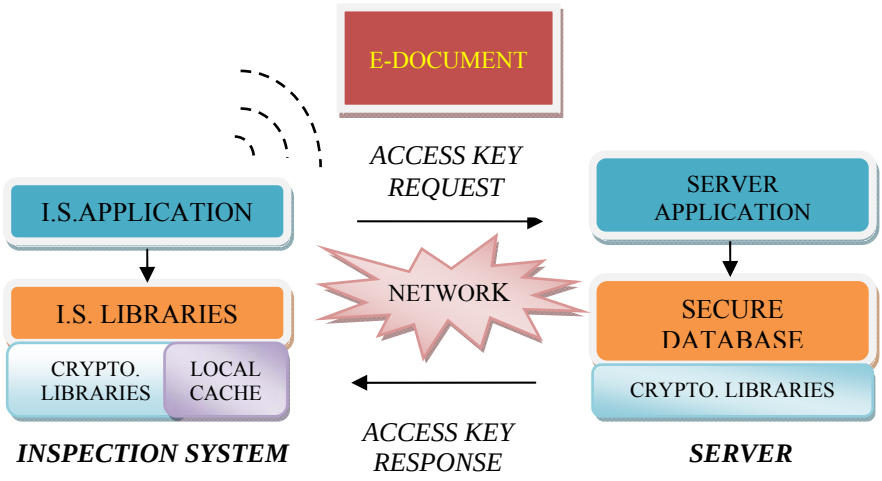

Figure 2: architecture of the access key infrastructure

The client module resides in the inspection system and receives key requests from the IS application. Once a key is requested, the client module authenticates itself to the server and establishes an encrypted communication channel. This encrypted channel is used to transfer the request from the IS application and receive the key from the server.

A local key cache is integrated in the client module. Thanks to this cache, it is possible to store previously used keys locally. When a key request is received, the local cache is checked first. If the key is located there, it is recovered directly without being authenticated to the remote server. Thus, this solution improves system availability in situations where we have no access to the server, and it provides the infrastructure with more execution speed and flexibility.

The server module mission is to authenticate any inspection system and check the security policies specified by the administrator before processing the request. The requests generated at the IS application are catalogued in two types: a new key request or a key retrieval request. New keys are generated at the server module according to the key parameters specified in the request and updated in the key storage. Key retrieval requests are processed against the storage module. Once the request has been processed, the appropriate key or error message is sent back to the IS client.

The storage module is responsible for storing each pair <key identification, key value> to make it possible for the server to recover any of the keys previously created if the client requests it.

There are some additional aspects relating to our solution that are worth discussion. First, this proposal focuses on a central-based infrastructure, that is, all the information required to generate and access the keys is stored in only one centralized server. This approach is better for systems where data must be stored in a highly secure node, since spreading the information in multiple nodes could compromise its

security and increase the latency to get the data if more than one node needed to be contacted for processing a request. On the other hand, a single central-server could turn into a bottleneck and a single point of failure if the number of requests exceeded server capacity. Although this approach could be enough in some cases, depending on the e-document type and its application scenario, a hierarchy of secure key servers could be necessary.

Secondly, back-up or alternate servers could also be used to avoid data loss due to network problems, in such a way that if one server is down, the client can communicate with a different one. The information on the alternative servers could be provided by the e-document itself, configured at the IS terminal or available in a third party server (e.g. a server which provides appropriate key server URLs depending on the e-document type or its characteristics). So, when the client requests a new key, it tries to connect to the first server on its list of servers. If it is available, the procedure continues till it gets the requested key. If the server is down, it tries to connect to the next server, and so on. Moreover, in some specific scenarios a complete or partial mirror key server could be available locally to avoid problems due to network availability issues. Which part of the access keys should be mirrored locally would also depend on the specific application scenario.

As far as the policies are concerned, the server allows configuring two different types: key policies and key cache policies. The first type makes it possible to set conditions for the use of the keys - e.g. the duration of a key before it expires in terms of minutes or the number of encryptions performed with it. It also allows specifying which algorithm to use, such as Triple-Data Encryption Standard (3-DES) or Advanced Encryption Standard (AES), as well as the size of the keys. As for the second type of policy, the key cache policies can specify whether we are interested or not in storing the keys in the cache, and if so, how long we want to keep them there. So, when the client receives a request from the IS application for a new key, first it looks in the local cache, and if there is a key stored there, it checks the key cache policies. If the policies specify that the key is still useful, the client takes it from the cache and sends it to the IS application. If not, the client requests a new key and (if the policy so defines it) stores it in the cache for future use.

6.2 Security Requirements

The infrastructure explained before has some security requirements that must be implemented in order to make it safe against attackers.

- Keys encryption: communication between the client requesting the key and the server must be encrypted to avoid an attacker from eavesdropping on the communication and violating data confidentiality.
- Storage integrity: it is important that the database records on the key storage be digitally signed before storage and verified upon retrieval to ensure their integrity has not been compromised. In this way it can be detected if an attacker has modified a record and the client can be warned accordingly.
- Storage confidentiality: database records on the key storage should be encrypted before storage to prevent an attacker who has gained access to the system from recovering any key.

- Server access control: it is required to establish an access control mechanism to the central server, in such a way that only authorized clients can ask for new keys or get keys previously generated and stored in the server. This could be achieved by means of security providers and the use of digital certificates to sign the requests from the clients and the responses from the servers.
- Cache keys integrity: as we have a cache which stores keys locally, we have to ensure that it is not possible for an attacker to either get these keys or to modify them. So, cached keys must be digitally signed and encrypted on storage as well as decrypted and verified upon retrieval in order to check their integrity.
- Authentication for administrative operations: as for the administrative operations in the server-side – e.g. modifying policies – it all the changes must be controlled and kept in a log for auditing purposes. It is also important to carry out a previous authentication process by means of digital certificates in order to avoid an unauthorized entity taking over the server.

6.3 Prototype Implementation

The previous concept of key management infrastructure which solves security issues exposed in BAC and would be suitable for general use e-document access control has been implemented in a prototype.

To build our prototype system and prove the feasibility of our approach, we have used two open-source initiatives: the Java Machine Readable Travel Document (JMRTD) and the Symkey projects. Depending on the requirements of a particular e-document implementation, this software could also be used as a base instead of starting the design from scratch based on the ideas presented in this paper.

The JMRTD project provides a free implementation of the Machine Readable Travel Document standards as defined by the ICAO. Therefore, this project provides a Java implementation of the ICAO security mechanisms, the memory organization and data fields of the e-Passports, a particular implementation of the MRTD standards. As the source code is provided, the specific data fields required in each particular e-document type can be defined and the security mechanisms can be modified. In this case, the Basic Access Control scheme was modified in order to test our proposal.

Both, the e-document internal application and the inspection system application were developed using this approach. The e-document application is a Java Card applet which requires JCOP cards. In our case, NXP SmartMX P5CT072 RFID cards were used on the e-document side.

The open-source Symkey project is an Enterprise Key Management Infrastructure (EKMI) developed by Strongauth, Inc. which objectives consist of providing high level security solutions to businesses through a Symmetric Key Management System (SKMS). We chose this software because it fulfils the security requirements previously described and its underlying architecture satisfies our necessities.

The SKMS consists of two parts:

- Symmetric Key Services (SKS) server: in charge of key generation, escrow and recovery. It contains all the encryption keys, as well as the Access Control Lists (ACLs) authorizing access to the keys. It allows defining

central policies for symmetric keys, such as the key lifetime or key size, and is capable of accepting SKMSL protocol requests.

- Symmetric Key Client Library (SKCL): a set of libraries which allows making requests to the SKS server, as well as encrypting and decrypting data according to the key policies, supporting 3DES, AES-128, AES-192 and AES-256. In addition, it makes it possible to perform SKMSL protocol requests.

Information interchange between the SKS server and the SKCL is carried out by means of SOAP (Simple Object Access Protocol), a standard protocol that defines the interaction and communication of the two entities by means of XML messages. SOAP technology is widely used since it provides a decentralized and extensible framework capable of working with any protocol stack.

The basic inspection system application consists of several tabs which provide different functionalities required in the e-document context: perform the security mechanisms (e.g. BAC, active authentication, passive authentication, etc), initialize an empty card or manage the e-document data (in our example, we have incorporated the optional biometrics defined for e-Passports). This inspection system application has been integrated with the key management infrastructure. In Figure 3, a screenshot of the IS application is shown following our alternative BAC scheme where the high-entropy K_{seed} is requested from the secure server and K_{ENC} and K_{MAC} are derived from it.

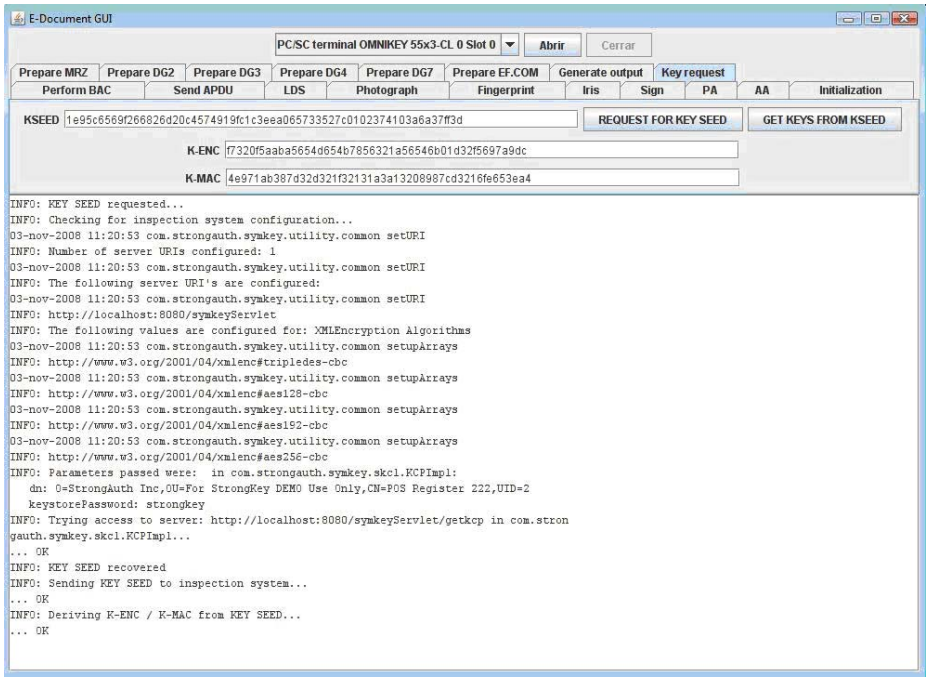

Figure 3: Screenshot of the IS application performing our alternative BAC scheme.

In our system, the IS application, instead of using the MRZ information to derive a static and low entropy K_{seed}, reads a key ID. After being authenticated to the secure key server, this key ID is used to recover the appropriate K_{seed} for the document (and to store it in the secure local cache according to the key policies). This K_{seed} is used to derive K_{ENC} and K_{MAC} and to complete the rest of the access control mechanism as in the original scheme. Thanks to this alternative scheme for generating and recovering K_{seed}, we obtain the maximum key entropy (in this case, 128 bits). The K_{seed} is independent of the e-document internal data and can be updated in each document inspection to prevent unauthorised (or grant-revoked) terminals from accessing the document information. Although we have used a document format and data fields similar to the e-Passport content for our application, an ad-hoc e-document format can be defined for each application scenario following this approach.

The open-source software initiatives used in the development of our e-document could also be adopted in the hybrid documents defined in some real application scenarios where development agility is required and the features provided fulfil their specific requirements. In other cases, a complete e-document redefinition and key management infrastructure design based on the ideas presented in this paper could be required.

7 Conclusions

Paper-based documentation is not likely to disappear in the near future as it is easy to use, convenient and reliable in daily life scenarios. However, traditional documents can go a step forward and share some of the benefits of digital e-documents. The integration of RFID technology in paper-based documentation can provide a seamless link between physical documents and the information system enabling novel features and potentially enhancing document security. The ICAO e-Passport is the most relevant 'hybrid document' until now, but others will undoubtedly come.

In this paper, we present our generalized concept of a hybrid document: a paper-based document enhanced with electronic features. We have presented the novel requirements that can be requested from an e-document in terms of natural integration with the digital information system and security features provided.

We have analyzed the weaknesses in the most comprehensive suite of security mechanisms available for e-documents (i.e. the ICAO security mechanisms for MRTDs) focusing on the adoption of their schemes in other types of hybrid documents. The Basic Access Control mechanism has some important weaknesses (namely, low key entropy and the use of a static key during the document lifetime) which can compromise the overall e-document security. Moreover, the key derivation mechanism used in the BAC scheme is not only the origin of these issues, but it is tied to specific e-Passport data fields, thus limiting the adoption of these mechanisms in other e-documents.

Finally, we have presented different alternatives to the original key derivation scheme in BAC in order to solve the low key entropy. In addition, we enable the use of a dynamic access key and do not tie the access key to specific e-document data fields. In particular a key management infrastructure for e-documents has been proposed and implemented in a prototype.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Industry through the IDENTICA project (FIT-360503-2007-3) and the Spanish Ministry of Science and Education through the research projects CRISIS (TIN2006-09242) and ARES (CSD2007-00004). The first author has been funded by the Spanish Ministry of Science and Education through the National F.P.U. We would like to thank our partner CITIC in the IDENTICA project for their support in the implementation of a personalized e-document using JMRTD.

References

[Lahiri 2005] Lahiri, S. RFID Sourcebook, Prentice Hall PTR, 2005.

[ICAO 2005] ICAO, Machine readable travel documents, Part-1, Machine Readable Passport Volume 2, Specifications for Electronically Enabled Passports with Biometric Identification Capability, Doc. 9303, 2005.

[Najera 2007] Najera, P., Lopez, J., RFID: Technological Issues and Privacy Concerns, In Digital Privacy: Theory, Technologies and Practices, A. Acquisti et al. (Eds.), Taylor and Francis, 2007.

[ISO 2000] ISO/IEC, ISO/IEC14443, identification cards – contactless integrated circuit(s) cards – proximity cards, 2000.

[Juels 2005] Juels, A., Molnar, D., Wagner, D., Security and privacy issues in e-passports, in 'IEEE Se-cureComm '05', 2005

[Jacobs 2006] Jacobs, B., Hoepman, J., Hubbers, E., Crossing borders: Security and privacy issues of the European e-Passport. IWSEC 2006, 2006.

[Matyáš 2007] Matyáš, V., Z. Říh., P. Švénda, "Security of Electronic Passports," CEPIS UPGRADE: The European Journal for the Informatics Professional (Vol. VIII, issue no. 6), December 2007, pp. 60 - 67.

[Vijayakrishnan 2008] Vijayakrishnan, P., Pieprzyk, J., Wang, H. "Formal security analysis of Australian e-passport implementation"'AISC '08: Proceedings of the sixth Australasian conference on Information security', Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 2008, pp. 75-82.

[NIST 2007] National Institute of Standards and Technology, Recommendation for Key Management, Special Publication 800-57 Part 1, NIST, March, 2007.

[ECRYPT 2008] Yearly Report on Algorithms and Keysizes, D.SPA.28 Rev. 1.1, IST-2002-507932 ECRYPT, July, 2008.

[Avoine 2008] Avoine, G., Kalach, K. and Quisquater, J. "ePassport: Securing International Contacts with Contactless Chips", in Tsudik, G., ed.,'Financial Cryptography and Data Security -- FC'08', IFCA, Springer-Verlag, Cozumel, Mexico, 2008.

[Carluccio 2006] Dario Carluccio, Kerstin Lemke-Rust, C. P. A. S. "E-passport: the global traceability or how to feel like an UPS package", Ecrypt, Printed handout of Workshop on RFID Security - RFIDSec 06, Graz, Austria, 2006.

[Oostdijk, 06] Laurie, A., Breunesse C., Poll E., Richter H., Hubbers E., Oostdijk, M., Ronny S., Mostowski W.: Java Machine Readable Travel Document (JMRTD) project: http://jmrtd.org/index.shtml

[Chakrabarti, 99] Chakrabarti, K., Mehrotra, S.: The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces, In Proc. Int. Conf. on Data Engineering, February 1999, 440-447 http://citeseer.nj.nec.com/chakrabarti99hybrid.html

[Cocoon, 02] Cocoon XML publishing framework, 2002, http://xml.apache.org/cocoon/

[Hunter, 00] Hunter, J.: Proposal for the Integration of DublinCore and MPEG-7, October 2000